NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MULTIPLE TARGET IDENTIFICATION
AND DIRECTION FINDING
USING MATCHED FILTERING TECHNIQUES

by

James L. Johnston

December 1983

Thesis Advisor: H. A. Titus

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Multiple Target Identification and Direction Finding Using Matched Filtering Techniques		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James L. Johnston		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE December 1983
		13. NUMBER OF PAGES 169
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

seismic sensors, matched filtering, target identification, target direction.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research investigates seismic signal processing techniques for battlefield target classification and acquisition. Multiple target classification is performed by discrete time domain matched filtering. Multiple target directions are determined using the responses of the matched filters and least mean squares polynomial curve fitting. The least mean squares polynomial curve fitting procedure is also used for direction finding for recoil/blast sources, using the unfiltered seismic signals.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Approved for public release; distribution unlimited.

Multiple Target Identification and Direction Finding Using Matched Filtering Techniques

by

James L. Johnston
Captain, United States Marine Corps
B.S.E.E., San Diego State University, 1975

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1983

ABSTRACT

This research investigates seismic signal processing techniques for battlefield target classification and acquisition. Multiple target classification is performed by discrete time domain matched filtering. Multiple target directions are determined using the responses of the matched filters and least mean squares polynomial curve fitting. The least mean squares polynomial curve fitting procedure is also used for direction finding for recoil/blast sources, using the unfiltered seismic signals.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

I wish to express my appreciation to Professor Harold A. Titus for his professional guidance and support in developing this thesis. I would also like to thank my wife, Pamela, for her constant support during the writing of this work.

I. INTRODUCTION

Timely and accurate combat intelligence is a integral part of the modern battlefield. A primary goal of combat intelligence is target acquisition. Hostile targets can be acquired by either passive or active means. An example of passive target acquisition is visual target identification. Radar, on the other hand, is an active target acquisition device. An effective combat intelligence system will include a mix of both active and passive target acquisition methods.

Rapidly advancing technology in the fields of electronic counter-measures and radiation-seeking weapons has enhanced interest in passive target acquisition methods. To be cost effective, as an additional target acquisition system, a passive system must be able to provide swift and accurate target identification, location and tracking information on hostile targets. A variety of seismic sensor systems have been used in this roll with varying degrees of sucess.

Naval Ocean System Command (NOSC) in San Diego has developed a system based on a circular ring of sensors with data collection managed by an array-processor/minicomputer system [Ref. 1]. The observation of enemy movements and activity beyond the Forward Edge of the Battle Area (FEBA) is the design objective of this system. This thesis uses data collected during a test of this system at the Marine Corps Air-Ground Combat Center at Twenty-nine Palms, Califiornia. Investigated are various methods of processing the seismic data collected. The objective of this research is to try to provide viable methods of satisfying system design objectives through signal processing techniques.

The following chapter addresses the design objectives and requirements for such a seismic sensor system. Additionally, capabilities and deficiencies of current systems and research are detailed. In order to intelligently address solutions to these requirements and deficiencies, an understanding of seismic theory and sensors is needed. General seismic theory is presented with emphasis on the constraining parameters for the use of the earth's surface as a medium for gathering seismic intelligence. Also highlighted are the similarities of the earth's surface to electro/optical phenomena and the resulting simplifying assumptions.

The sequence of design solutions investigated followed from the analogies and simplifying assumptions addressed in the study of seismic sensor theory. The problem of target identification or classification is approached using digital matched filtering of the time domain amplitude data. Frequency domain matched filtering was not considered, based upon the conclusion by NOSC that there appeared to be no consistent spectral lines for any of the possible target types, except for artillery [Ref. 1]. Matched filtering was used to identify single and multiple target classes occuring during a sample period. The chapter detailing the matched filter procedures and implementation also includes a description of support and validiation software used in the analysis of the seismic data.

The validation software is primarily used to check the accuracy of the direction finding routines. These direction finding routines are the time domain phase difference procedure (TDPD) and a least mean squares polynomial (LMSP) curve fitting procedure. The combining of the matched filtering procedure and these direction routines allowed for multiple target direction finding. The theory and derivations of the two direction finding algorithms are presented in the multiple target direction chapter.

Application of these algorithms is performed first on simulated targets for validation of the procedures. The experimental data is then analyzed. A user's manual, which includes procedures for tape and mass storage operations, is provided as an appendix. This appendix describes how to set up and use the software system.

II. THEORY OF SEISMIC SENSORS

Elastic waves result from the stressing of an elastic media. The elastic media for seismic theory is the earth. Seismic theory is the study of the earth as a wave propagating media. Elastic waves propagate away from the source of seismic stress, e.g. an explosion [Ref. 2]. The energy, which propagates through the earth, travels via particle deformations. The elastic properties and densities of the earth media determine the velocities of these seismic waves. [Ref. 3]. Seismic wave sources of interest may be impulsive or continuous. Impulsive or short duration sources are artillery recoil or shell blasts. The time-limited nature of this type of seismic signal produces a broad range of frequencies. Continuous wave signals may be produced by tanks, trucks and low flying aircraft. These continuous wave signals may be described by narrow band frequency characteristics. The spectral power of a seismic source is a function of several parameters. A non-inclusive list of these parameters includes:

1. The vehicle's velocity and mass
2. The size of the explosive charge associated with the artillery or shell blast
3. The degree of coupling into the earth's surface
4. The geological structure over the wave's path

Information about the seismic source is contained in the waves which it generates. For example, in an array of seismic sensors, directional information is contained in the relative received signal phases. In otherwords, the relative phase differences between the signals received by the individual array elements can be used to compute the direction to the seismic source [Ref. 4]. These relative phase

differences represent the time delays of the waves as they pass the array's elements. The response of the array's elements is proporticnal to the amplitude and velocity of the earth's motion, relative to the geophone's sensing axis for the waves. [Ref. 3]

Assumptions about the propagation of seismic waves must be made to assist with their analysis. The earth is assumed to be made up of horizontal, homogeneous, and isotropic layers of material. These layers are assumed to be discontinuous in their elastic properties at their borders. This variance in the elasticity between the layers leads to an optical analogy for the wave propagation. Propagation paths may now be viewed as being direct, refracted, or reflected versions of the source's seismic waves. [Ref. 3]

There are four basic types of seismic waves. These types are compressional, shear, Love, and Rayleigh waves. Compressional waves are generated by impulsive sources such as shell blasts. Particle motion is along the direction of travel. Shear waves are characterized by particle motion orthogonal to the direction of motion. The Love wave is a surface wave which may occur as a result of the layering of the earth's surface. This layering effect acts as a wave guide for this type of wave. Particle motion is orthogonal to the direction of wave propagation. The Rayleigh wave is generally the strongest of the seismic waves. The Rayleigh wave travels along the free surface of the earth. Its particle motion direction is always in the vertical direction. It is the strongest wave generated by a compressional source. Its amplitude attenuates at a rate only inversly proportional to the square root of the distance. [Ref. 3]

The waves of primary interest for a seismic system are the Rayleigh and Love waves. This is due to the long-range propagation of these waves. Since these two wave types are orthogonal to each other, they must be sensed by different

geophones. Rayleigh waves may be sensed by vertical geophones and the Love waves by horizontal geophones. Since the Rayleigh wave is normally the strongest, and in order to reduce computational complexity, only verticle sensor data is used.

Rayleigh waves experience absorbtion losses, particularly at higher frequencies. This phenomenon occurs because of the lowpass filter effect of the earth. This filtering effect is further compounded due to the fact that the cutoff frequency of the earth diminishes with range. Further complications arise due to the dependence of wave velocity on frequency. The result being that the wave train may change with distance, reducing the correlation of the wave shape between its source and distant points. Other sources of error occur because of the weathering of the surface layer, irregularities in the sub-surface composition, variances in the earth's layers and surface geometry. [Ref. 3]

III. DEVELOPMENTAL REQUIREMENTS AND PROBLEM DEFINITION

A. PASSIVE TARGET ACQUISITION AND SURVEILLANCE

A battlefield commander possesses a definte requirement for real time combat intelligence. A significant tactical advantage is held by the commander who is able to integrate his availiable combat intelligence sources with his supporting arms, i.e., target acquisition and engagement. It follows that to be part of the target acquisition process, any real time, seismic sensing system must provide swift and accurate information on detected enemy targets. The specific requirements for such a system are the ability to detect, identify, and locate these targets [Ref. 5]. Additionally, the target's rate and direction of movement should be provided or made easily discernible.

Any seismic sensing system must be designed around the target acquisistion cycle. The target acquisition cycle, as given by Dublin [Ref. 5], is as follows:

1. Search Time
2. Target Sensing
3. Information Processing
4. Display of Target Information
5. Analysis of Target Information
6. Time required to make a Decision
7. Time Required for Supporting Weapons to Respond

For a seismic system, a prioritized list of possible targets are as follows:

1. Artillery
2. Helicopters and Aircraft
3. Tracked and Wheeled Vehicles
4. Personnel

As may be expected, the relative amplitudes of these seismic targets vary widely. A seismic targeting system is therefore constrained as to the targets it can or can not be expected to effectively engage.

The variance in the relative amplitudes of seismic targets suggests a range of specifications for detection of these targets. As summarized by Dublin [Ref. 5], possible detection radii may be as shown in Table I. Radii are given for both short and long targeting systems.

TABLE I

Target Detection Radii

Target	Short Range System	Long Range System
Personnel	100M	None
Vehicles	1KM	10-20KM
Low Flying Aircraft	1KM	10-20KM
Hostile Weapons	1KM	15-20KM

The timeliness requirement is ancillary to the radii of detection specifications. Timeliness, as used here, refers to the total time commencing when the seismic sensor system first detects and processes the seismic target data and ending with the dissemination of the targeting information to command elements for disposition. This timeliness requirement ranges between five to fifteen minutes, depending upon the mobility of the target [Ref. 5].

The parameters having a direct effect on the timeliness of a system are the probabilities of false alarm and detection. These parameters directly relate to a system's value. Increased probability of detection with reasonable false alarm performance, combined with the ability to disreguard

friendly targets, are practical design objectives for any targeting system. The ability to incorporate such design features into a seismic sensor system will reduce both the time wasted on invalid targets and the danger of undetected targets.

Once a valid target has been detected, target location information must be obtained. Stringent specifications for target locations allow for the system to support or enhance the effectiveness of; fire support systems, blind bombing, Harassing and Interdiction fire (H and I), and observerless artillery engagement.

B. CURRENT CAPABILITIES AND DEFICIENCIES

To date, numerous successful algorithms have been developed to determine direction to single targets. These algorithms include both time and frequency domain methods. Target identification of long range targets via seismic sensing has not yet met with equal success.

The modern battlefield is seldom a single target type environment. The complicated, real world problems of multiple target identification and multiple target engagement require solution before practical seismic sensor systems can be integrated into the target acquisition process.

IV. MATCHED FILTER CONCEPT AND DISCRETE ALGORITHM

A. MATCHED FILTER THEORY

As previously addressed, there exists a requirement for battlefield target identification/classification. The recovery and classification of target signals suggests a filtering requirement. Previous works and implementations have used frequency domain techniques [Ref. 4]. The Air Force's SKEET system and the U.S Army's Remote Battlefield Surveillance System (REMBASS) both have successfully implemented a spectral power approach for classification of seismic data. These systems, however, are for short range applications. Time domain approaches to target identification have been for the most part left unexplored.

The discrete matched filtering technique is an attempt to classify targets by their time domain amplitude pattern i.e., their seismic amplitude signature. The heuristic basis for this method evolved from the observation of visual differences between the amplitude versus time signals for the various classes of targets. The matched filter, being the optimum filter for detecting known signals, was selected [Ref. 6].

The discrete matched filter is described by equation 4.1, where $h(t)$ is the impulse response of a filter whose output signal to noise ratio at time t_0 is maximized. The unit step $u(t)$ has been added to assure causality for the system. The matched filter is [Ref. 7].

$$h(t) = s(t_0 - t)u(t) \tag{4.1}$$

The output signal is given by

$$sout = \int_{-\infty}^{\infty} h(v)\,s(t - v)\,dv \qquad (4.2)$$

The maximum signal to noise ratio is given by

$$(sout^2/M^2)maxoutput = E(t_o)/N_o \qquad (4.3)$$

Where M is the noise level at the filter output, N is the input noise level, and $E(t_o)$ is the energy in s(t) up to time t_o. In equation 4.2, the replica of the original known signal is reversed and translated in time to be convolved with the signal input to the filter, producing the optimum output signal to noise ratio.

For discrete realization of the matched filter as implemented, equation 4.2 becomes equation 4.4, where h(k) is the reflected and translated known signal.

$$sout(j) = \sum_{k=1}^{N} h(k)\,s(j - k) \qquad (4.4)$$

Where N is the number of data points per sample period.

B. MATCHED FILTER ALGORITHM

Samples of known signals are stored in a data file and read into a 5120 array at the start of program execution. Five sample, or known filter signals, are recorded in this array. Each sample signal comprises 1024 of the 5120

elements of the matched filter array. To perform the matched filtering, the 1024 seismic data samples (unknown signal) are copied over five times on an array of 11264 size. These copies of the experimental data are separated by 1024 zeros on either side. Additionally, the leading and trailing elements of the experimental data are set to zero to eliminate switching spikes present in the data. Hence forth, this 11264 element array will be referred to as the working array for brevity. This working array will contain the results of the matched filtering. For M, given as the number of input signal and filter signal array elements and also the number of zeros, the requirement set forth in [Ref. 8] for nonoverlapping convolution of length L is satisfied. Equation 4.5 establishes the minimum length for nonoverlapping convolution of one matched filter segment.

$$L = 2M - 1 \qquad (4.5)$$

Figure 4.1 depicts the working array layout. Notice that the first filter signal, "Tracked Vehicle" is loaded into the h(t) array and is convoluted with the working array elements one through 2048. Once the leading data point of h(t) reaches the working array's element 2048, a new known signal is loaded into h(t), "Wheeled Vehicle", and the convolution is continued for working array elements 2048 to 4096. This process continues until the last of the five h(t) filter signals has been read in and the convolution has been performed on all copies of the data. Notice that working data array elements 10240 to 11264 are for array length over-run protection.

Prior to convolution, each filter sample signal and the input signal are equalized to the same power level through

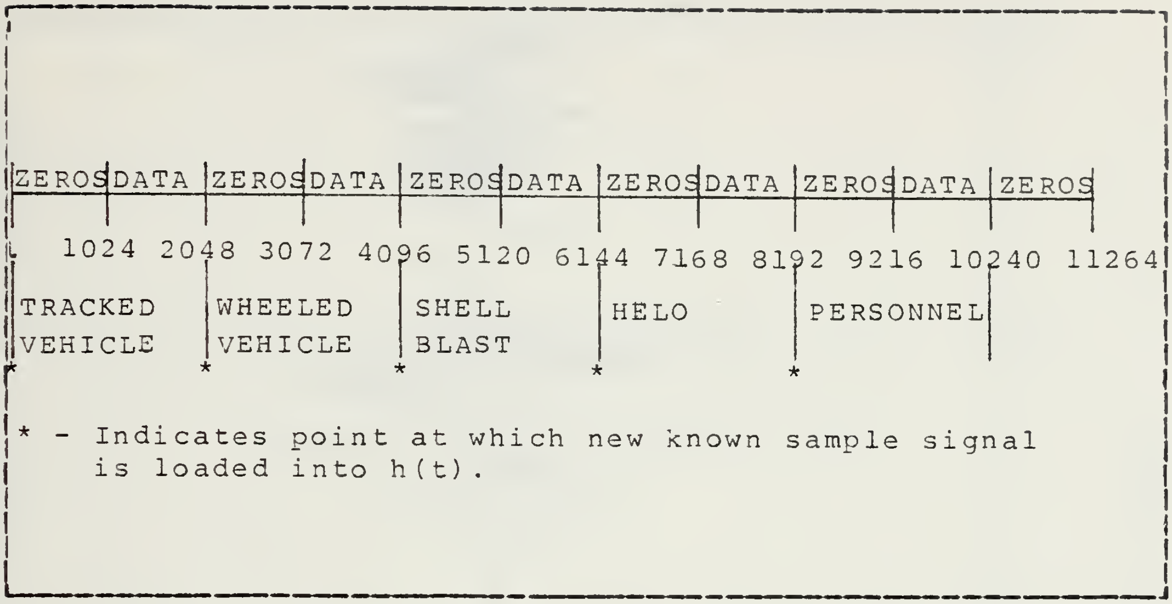

Figure 4.1 Working Array Configuration

division by their respective root mean square values. Additionally, after the convolution, the entire working array is normalized with respect to its maximum amplitude element. The maximum value in each target classification section is then compared with the interactively selected matched filter threshold. If the section's peak value is above the threshold value, that class of target is declared to be present. This method allows for the simultaneous detection and classificaton of multiple targets. It should be noted that this equalization forces both high and low amplitude signals to an equivalent average power level. This is felt to be justified in that signal amplification is not the goal, rather target detection and classification is. In the case of a single target, the known signal that most closely matches the unknown signal is anticipated to have the greatest amplitude matched filter spike. Figure 4.2 is a sample output.

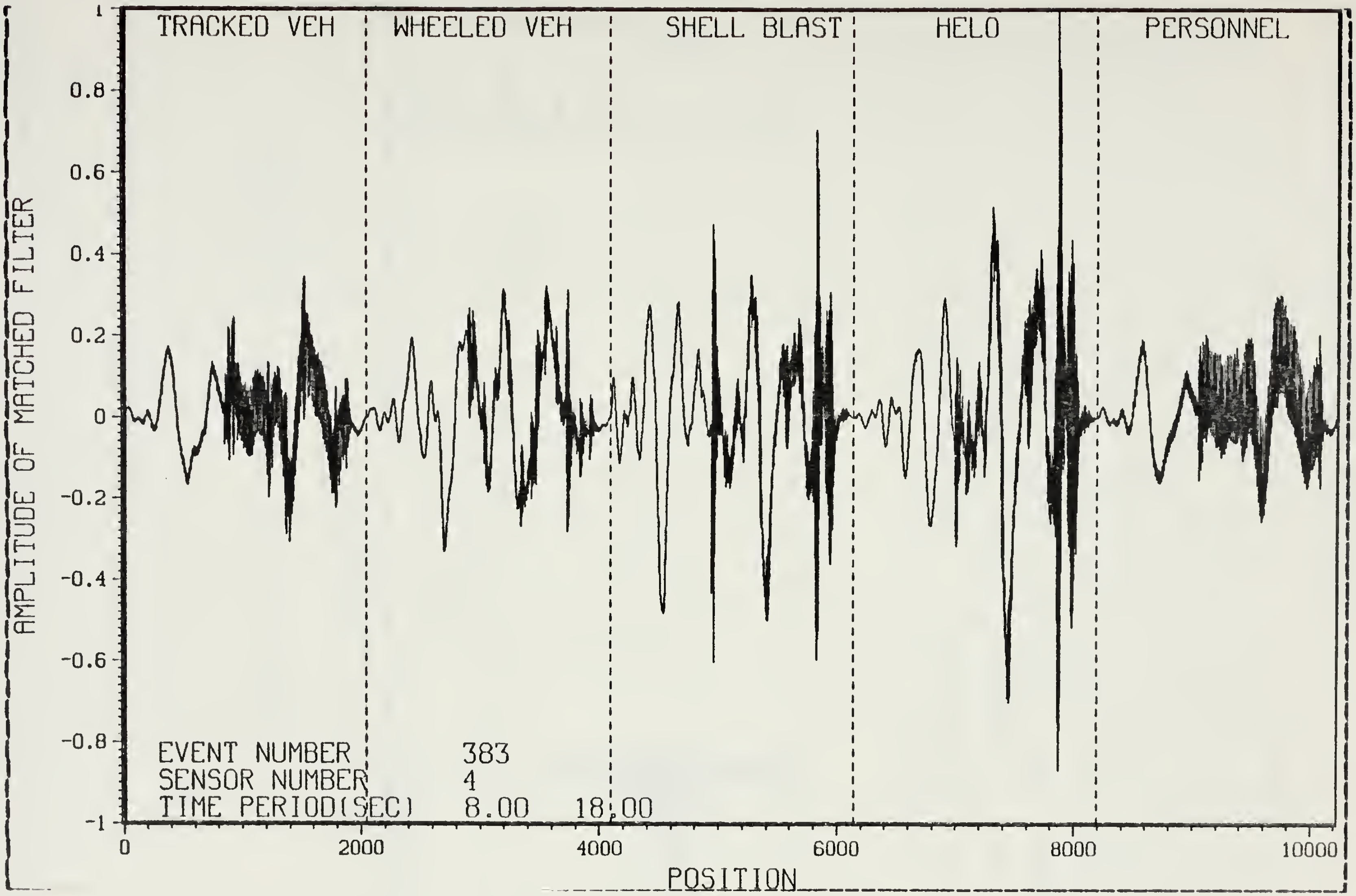

Figure 4.2 Sample Matched Filter Output

C. MULTIPLE TARGET MATCHED FILTERING

The upper limit of equation 4.4 may be selected to be from one to 1024 when called by the multiple target direction routine. This allows for the selection of reduced program execution times. Target identification, however, is always made by the full 1024 element buffer. When the matched filter target identification routine is used by the multiple target direction finding routine, data windows of less than 1024 are formed from a segment of the 1024 elements by extracting the segment size required around the maximum signal value. Figure 4.3 illustrates this procedure.

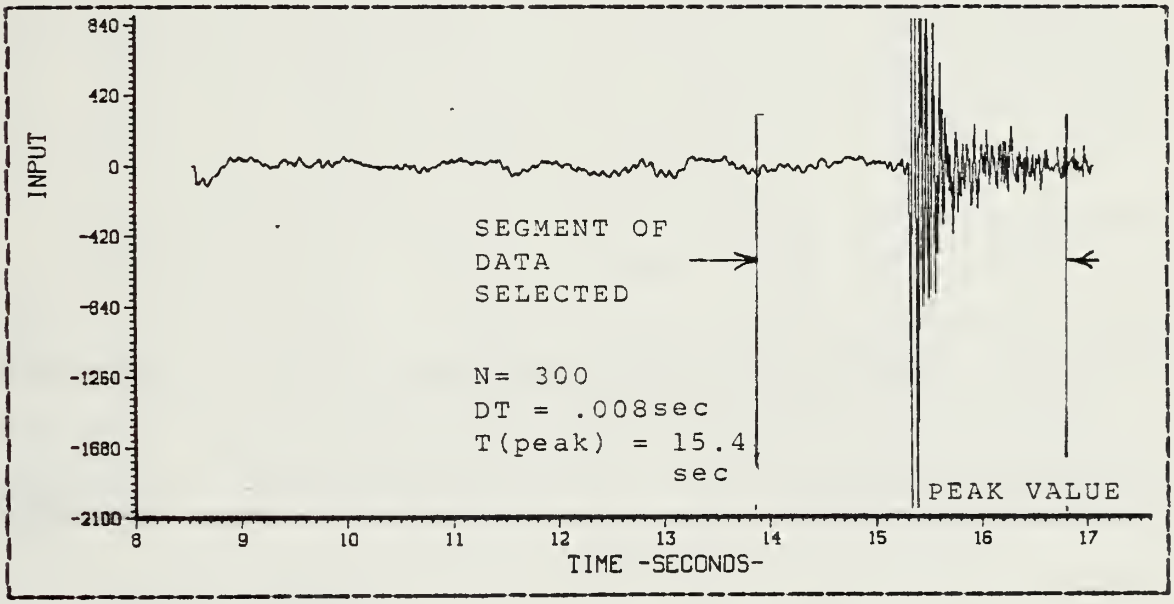

Figure 4.3 Windowing of Experimental Data

D. COMPUTATIONAL REQUIREMENTS

The computational requirements for this procedure are described by equation 4.6, where NO is the total number of operations required by the matched filter routine.

$$NO = (Ndf^2)(Nc) + (Ndf - 1)(Ndf)(Nc) \qquad (4.6)$$

Ndf is the number of data elements and also the number of filter elements. Nc is the total number of target classes. The term Ndf^2Nc is the total number of multiplications required, while $(Ndf - 1)NdfNc$ is the total number of additions.

When used for target identification, Ndf equals 1024 and Nc equals five. Equation 4.6 gives a total of 10,480,640 operations for these array sizes. When the matched filter routine is initiated by the multiple target direction routine, the value of Ndf can be selected to range from one to 1024. Figure 4.4 is a plot of equation 4.6 and shows the computational consequence for selection of large values for Ndf. Note that equation 4.6 must be multipled by the number of sensors in the ring when computing the number of operations for the multiple direction routine.

Window sizes of 100 and 200 were found to provide acceptable accuracy with greatly reduced computation times. As shown by figure 4.4, a window size of 200 requires 399,000 operations, while a window size of 100 requires only 99,500 operations. The number of operations required was found, as expected, to be proportional to the execution time of this routine.

E. SUPPORT SOFTWARE

1. Amplitude Analysis (Timout)

A graphical output is provided by this amplitude routine which displays the relative amplitude versus time for a selected sensor. The initial target direction and target classifications found are also displayed. This routine allows for the interactive selection of the

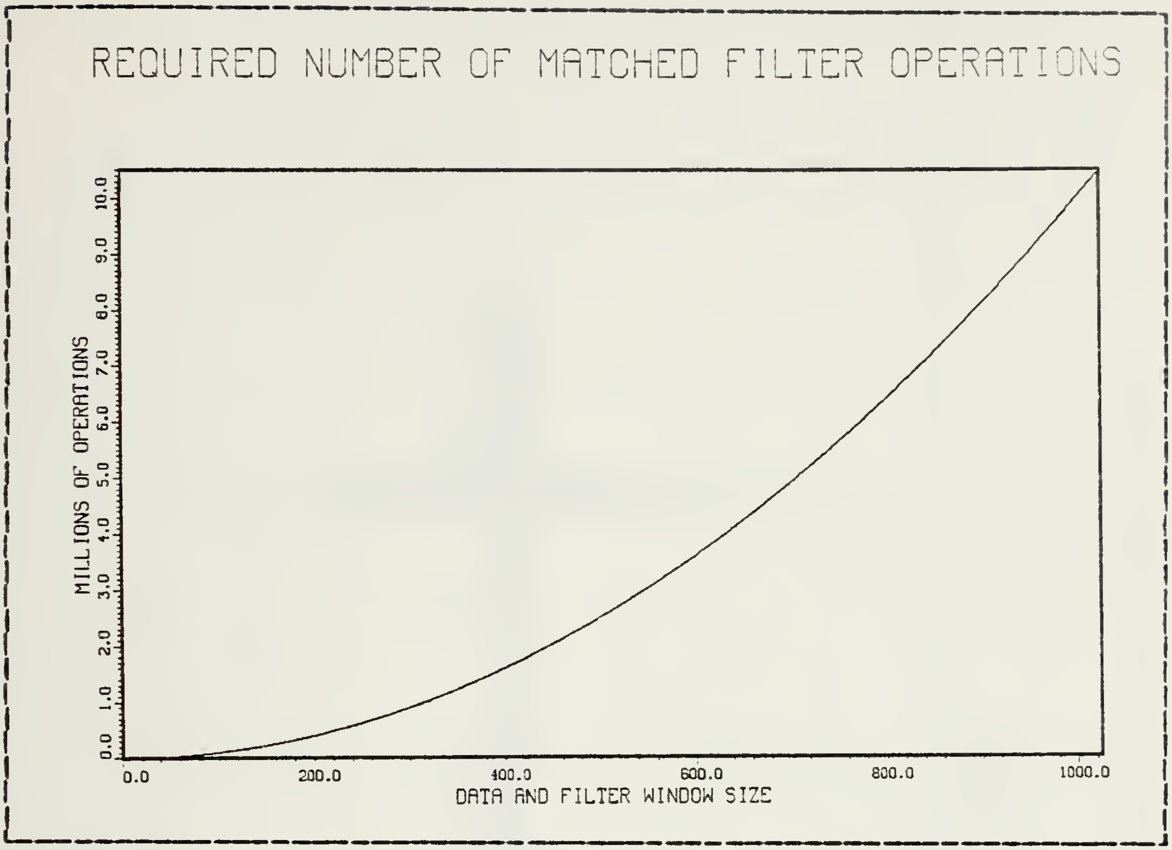

Figure 4.4 Data Window Size vs Number of Operations

amplitude response of any sensor as a sample target for later use in the matched filter analysis. In this way sample, signals can be catalogued and evaluated as filter signals. The axes of the graphical output adapts to the data's maximum amplitude and to the time period involved. Figure 4.5 is an example of amplitude analysis output.

2. Frequency Analysis (Freqot)

The frequency routine, as in the amplitude analysis routine, allows for initial primary target direction, target classifications, and for the time period of the experimental data to be displayed. Normalized spectral power versus frequency is graphically displayed as shown in figure 4.6.

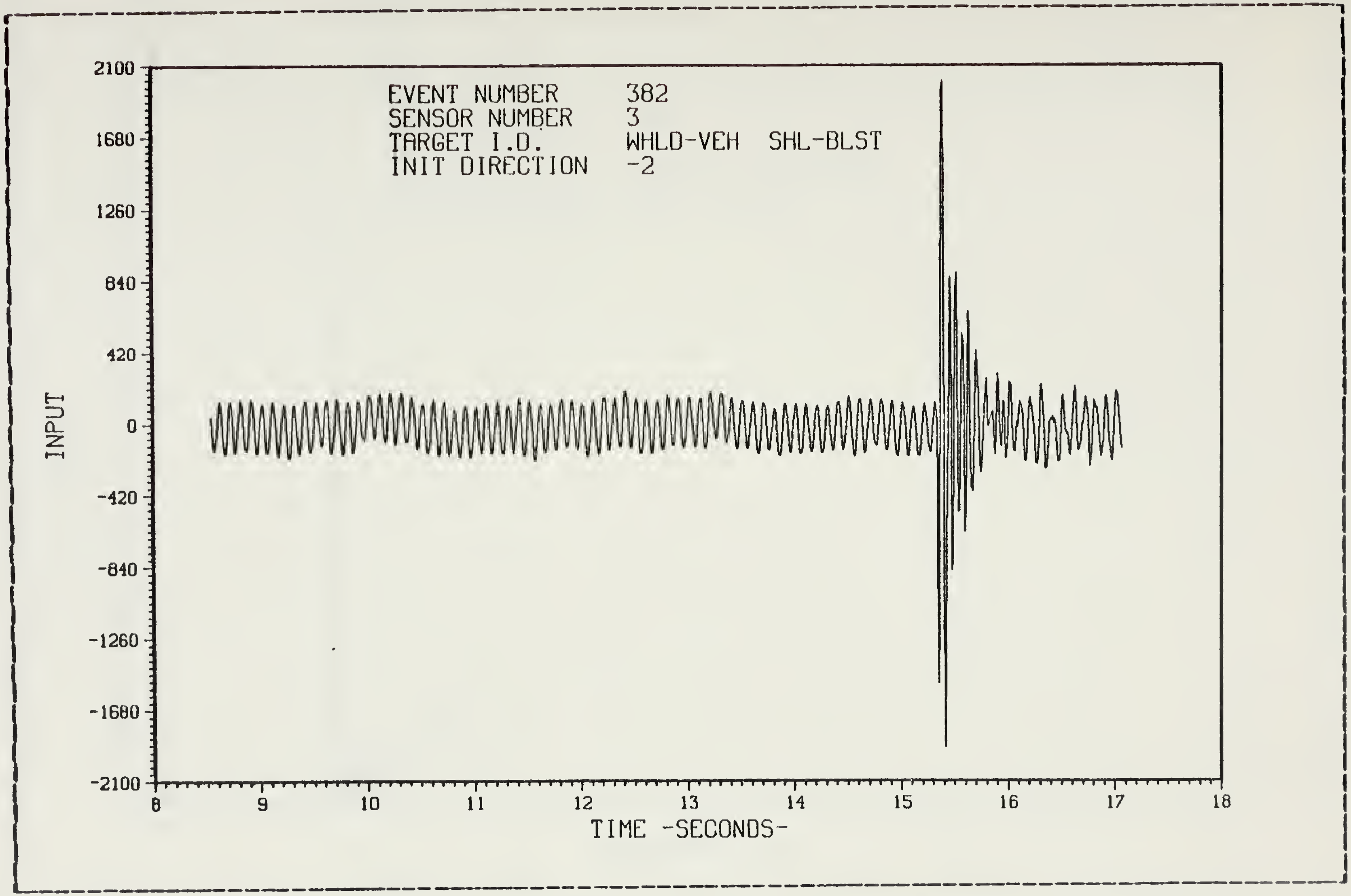

Figure 4.5 Sample Amplitude versus Time Output

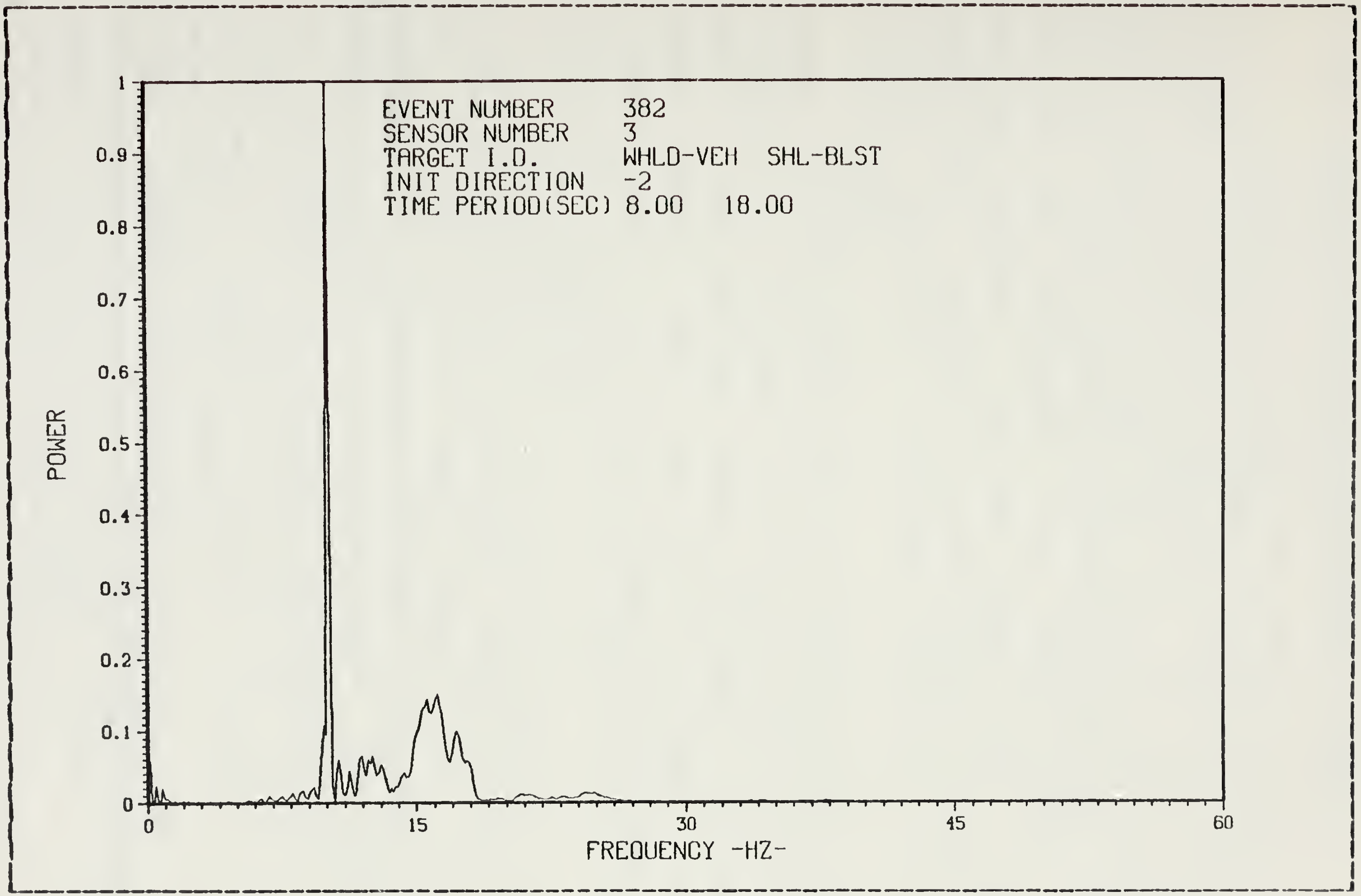

Figure 4.6 Sample Frequency versus Power Output

3. Simulation and Validation Software (SIMULT)

In order to validate the various algorithms and their implementation in software, a testing procedure was required. The specific algorithms which the simulation routine was designed to validate are the initial angle, phase difference and least mean square target direction routines. These routines, as previously described, use the relative time differences of the seismic signal's peak amplitude response or the peak matched filter response respectively. Validation of these routines is performed by allowing the creation of simulated targets with selected arrival angles.

Simulated targets can be created in the routine SIMULT. Up to four sine wave targets of selected frequency, amplitude, and direction can be input during each sample period. These simulated targets are added to the experimental seismic signal data for the sample period. Correspondingly, zero direction filter data must have been written into the matched filter data file at these selected sample target frequencies.

The directions for the simulated targets are created by introducing relative phase delays between the sine waves that are added to each sensor's seismic data. This is implemented by introducing a zero phase to the sensor in the desired direction of the simulated target. The phases of the other sensors are increased proportionally by their distance on the circular array away from the zero phase shift sensor. Figure 4.7 illustrates the case of a zero degree simulated target direction.

A summary of the target directions found by the multiple target direction routine and the simulated targets entered is provided by the multiple direction plotting routine. Figure 4.8 is a sample output.

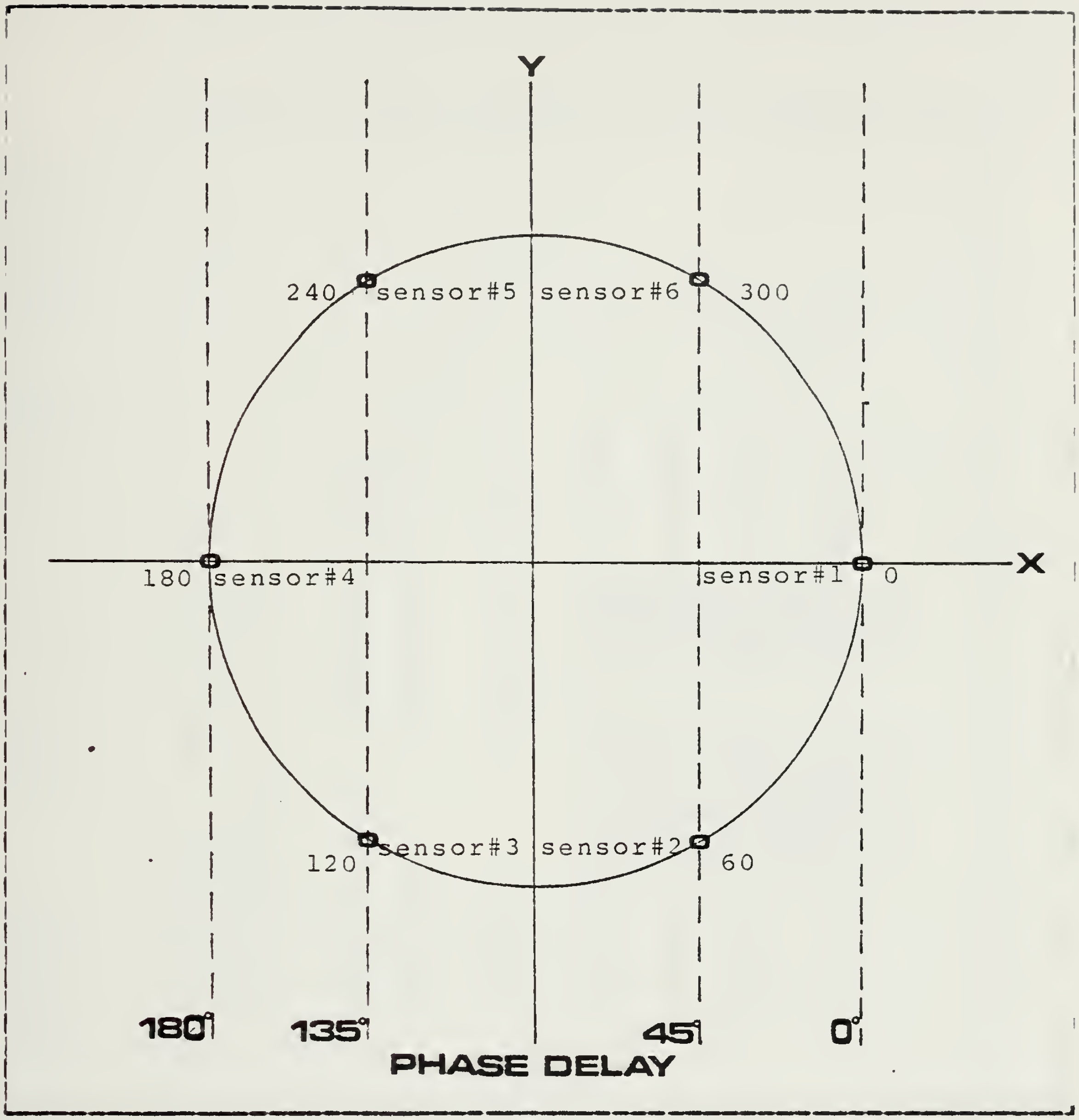

Figure 4.7   Simulation of a Zero Degree Target

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```
EVENT NUMBER     375

TIME PERIOD(SEC) 25.00  35.00


WHEELED VEHICLE   DIRECTION --59.00

SHELL BLAST       DIRECTION - 315.00


PERSONNEL         DIRECTION --59.00


SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 4.8   Sample Multiple Target and Simulated Target Output

V. MULTIPLE TARGET DIRECTION

A. THEORY AND DESIGN CRITERION

The modern battlefield is comprised of many classes and quantities of seismic signals. Using these signals for target identification and acquisition is the purpose of battlefield seismic sensors. As presented in the last chapter, matched filters can be used for target identification. In this chapter, it will be shown that matched filter information may also be used to obtain target bearing.

Multiple target acquisition requires the concurrent separation of the target classes and the computation of direction for each of the target classes found. The output of the matched filter is a spike at t_0, the time of peak signal detection. If matched filtering is performed for each of the sensors in the ring, the value of their respective t_0, for each filter output, would be different. Since the seismic waves impinge upon each sensor at different times, dependent upon the target direction, the values for t_0 for each sensor may be expected to be directly related to the arrival angle of the seismic wave. The t_0 spike of the matched filter output may be thought of as a signal compression for both the continuous (tank, truck etc.) and the time limited (shell blast, artillery recoil etc.) seismic signals [Ref. 6].

Time domain methods [Ref. 1] use the positional differences of known wave points to geometrically estimate direction. The times associated with a sensor ring's peak amplitude responses may be explained by way of an illustrative example of a shell blast. A rough direction to the origin of this shell blast can be computed using the

relative time differences associated with the peak amplitudes of all of the ring's sensors [Ref. 9].

The enhanced time positional information, which is a by-product of the matched filtering, can be used to perform just such a time domain approach. The motivation for the method is that, unlike other time and frequency domain methods, which are very susceptible to noise corruption inaccuracies, matched filtering pulls the signal out of the noise and optimally detects the signal at time t_o. Two time domain methods are evaluated for finding arrival angles of the seismic signal. The first being the time domain phase difference (TDPD) method. The second being a least mean squares polynomial (LMSP) curve fitting approach.

B. MULTIPLE TARGET FILTERING ALGORITHM (MULTI)

The routine MULTI calls the matched filter routine for each sensor's amplitude signal. Returned are the target classes found with their relative peak filter response positions. Figure 5.1 illustrates a two target case. The matched filter response and the relative time postion for the two classes of targets can be seen. The figure shows that a shell blast target is present. The relative time for this target class is 5800. The relative time returned for the simulated wheeled vehicle target is 3000. These times or positions are relative since each sensor's filter response peaks are offset in time with respect to the peaks of the other sensors. This allows for simultaneous direction finding for each class of target. This algorithm allows selection of either a time domain phase difference or or a least means square polynomial algorithm for finding the direction to the targets. The time domain phase difference algorithm is derived first [Ref. 1]. The least mean squares polyncmial direction algorithm then follows and is believed to be an orignial application to this field [Ref. 10].

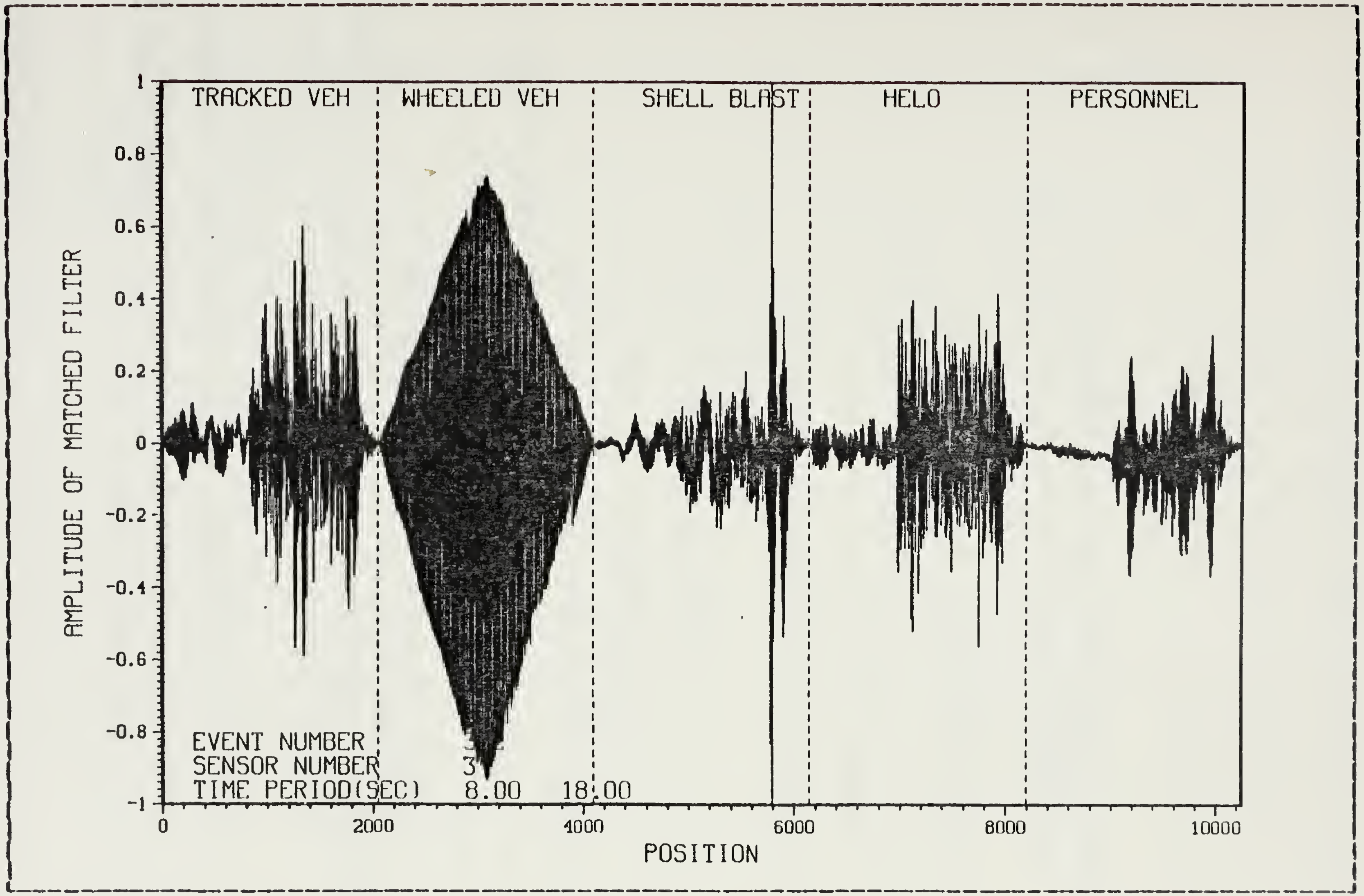

Figure 5.1 Two Target Matched Filter Response

1. Multiple Target Direction Phase Difference Algorithm

T_{ij} - the arrival time of the seismic signal at the I-th sensor for the J-th target class

θ_i - the angle of the I-th sensor to the x - axis

D_{ji} - the distance from the origin to where the wave front of the J-th target passes the I-th sensor

Tc_j - the time when the J-th target class passes the orgin

X_i, Y_i - the position of the I-th sensor

B_j - the arrival angle of the seismic wave for the J-th target class

V - the seismic wave velocity

R - the radius of the sensor ring

I - the sensor number where I has integer values form one to nine

J - the J-th target class

N - the number of sensors in the ring

Figure 5.2 illustrates these parameters and their interdependence. The derivation of the algorithm follows:

$\theta_i = 2 (I - 1)/N$

where zero degrees is set parrallel to the x - axis

$X_i = R\cos\theta_i$

$Y_i = R\sin\theta_i$

$D_{ji} = R\cos(\theta_i - B_j)$

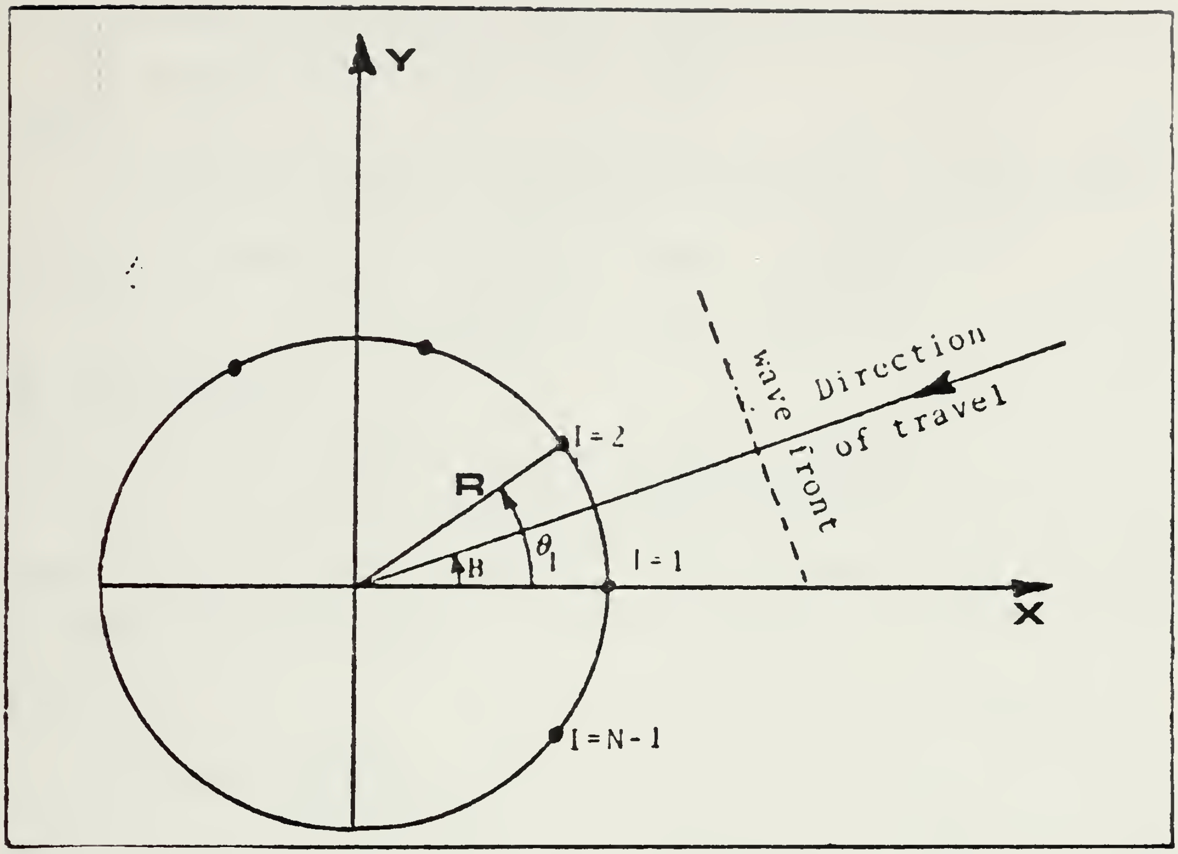

Figure 5.2 Circular Sensor Array Geometry

or equivalently $D_{ji} = (X_i)\cos B_j + (Y_i)\cos B_j$

$$Tc_j = (1/N)\sum_{i=1}^{N} T_{ji}$$

$$\tau_{ji} = Tc_j - T_{ji}$$

$$V = D_{ji}/\tau_{ji}$$

$$V = ((X_i)\cos B_j + (Y_i)\sin B_j)/(Tc_j - T_{ji}) \tag{5.1}$$

Since the wave velocity can be assumed to be constant when passing all sensors, then for any sensor I and K where $I \neq K$, equation 5.1 leads to

$$((X_i)\cos B_j + (Y_i)\sin B_j)/(Tc_j - T_{ji}) =$$
$$((X_i)\cos B_j + (Y_i)\sin B_j)/(Tc_j - T_{jk})$$

Where $I \neq K$

Now solving for the arrival angle B_{jik} of the wave

$$B_{jik} = \arctan(((Tc_j - T_{jk})X_i - (Tc_j - T_{ji})X_k)/ ((Tc_j - T_{ji})Y_k - (Tc_j - T_{jk})Y_i))$$

or equivalently;

$$B_{jik} = \arctan(((Tc_j - T_{jk})\cos\theta_i - (Tc_j - T_{ji})\cos\theta_k)/ ((Tc_j - T_{ji})\sin\theta_k - (Tc_j - T_{jk})\sin\theta_i))$$

Where the values of T_{ji} and T_{jk} are returned values from the matched filter routine.

Now;

$$B_j = (1/(N)^2)\sum_{i=1}^{N}\sum_{k=1}^{N} B_{jik}$$

Where B_j is the direction in radians to the J-th class target. For the multiple direction routine as implemented, 'j' has values from one to five.

2. Least Mean Square Polynomial Direction Finding

The least mean square direction finding algorithm was developed in response to problems encountered with the phase difference direction finding algorithm. This new method is base on a least mean squares polynomial curve fit of the sensor data. This approach was selected since the least mean squares polynomial provides for best fit or a maximum likelihood curve fit for noisy data.

The least mean squares direction finding algorithm, as with the phase difference algorithm, assumes the seismic wave to be planar. Figure 5.2 illustrates the parameters for this model. Once the assumption of a plannar seismic wave is made, the expected relation between the arrival

angle, relative delay times and sensor positon in the circular array, can be made. Figure 5.3 illustrates these relations for a nine sensor circular array with a seismic wave arriving at zero degrees. Notice that the relative delay times have been scaled to be from zero to one.

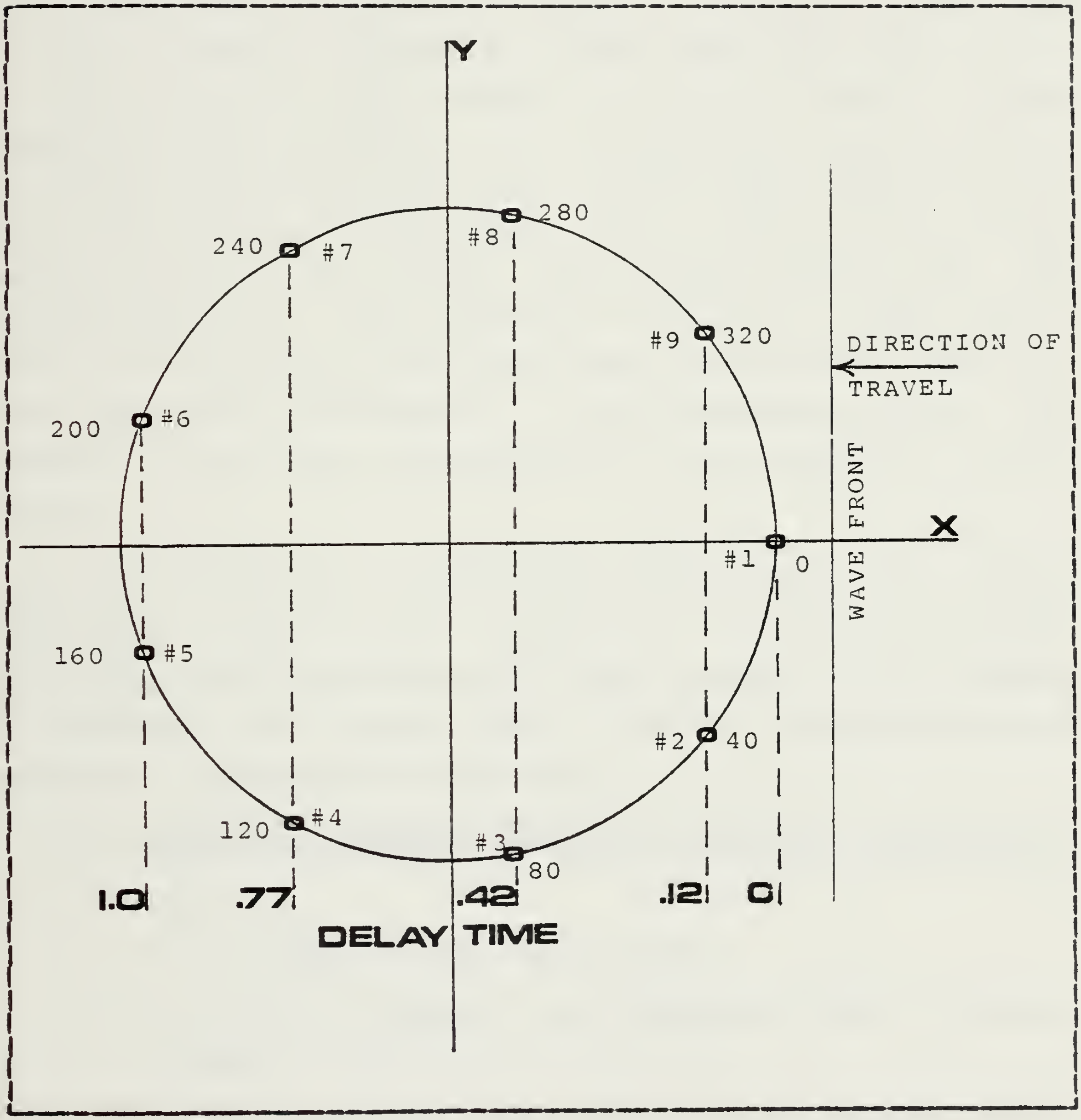

Figure 5.3 Relative Delay Times in a Nine Sensor Ring

Since the sensors in a nine sensor ring are at increments of forty degrees relative to the x-axis, a correlation can be seen between the relative time delays at each sensor and the arrival angle of the seismic wave. A plot can now be made to illustrate the relationship of sensor angles versus relative time delays. Figure 5.3 is a plot for a nine sensor ring with a plannar seismic wave arriving at zero degrees. Figure 5.4 shows that for a wave at an arrival angle of 160 degrees, the delay time at sensor number five will be zero. It can be seen that the fitting of these ideal data points with a least mean squares polynomial will produce an equation for a curve whose minimum value is also at the arrival angle of the seismic wave. The minimum degree of the polynomial to fit this ideal data is four. This results from noting that the curve in figure 5.4 has three curve inflections. For experimental data, this minimum curve point corresponds to the predicted arrival angle.

Polynomials of degrees higher than four may be expected to enhance arrival angle errors since the polynomial would distort to fit noisy data. Least means squares polynomials of degree two and three however, may be useful in reducing curve sensitivity to one or two malfunctioning sensors or excessively noisy data.

3. Least Mean Squares Polynomial Algorithm Derivation

Let:

N - number of data point pairs

Y_i - the observed or experimatal data position values

X_i - independent degree values with a range of 0 to 360 degrees

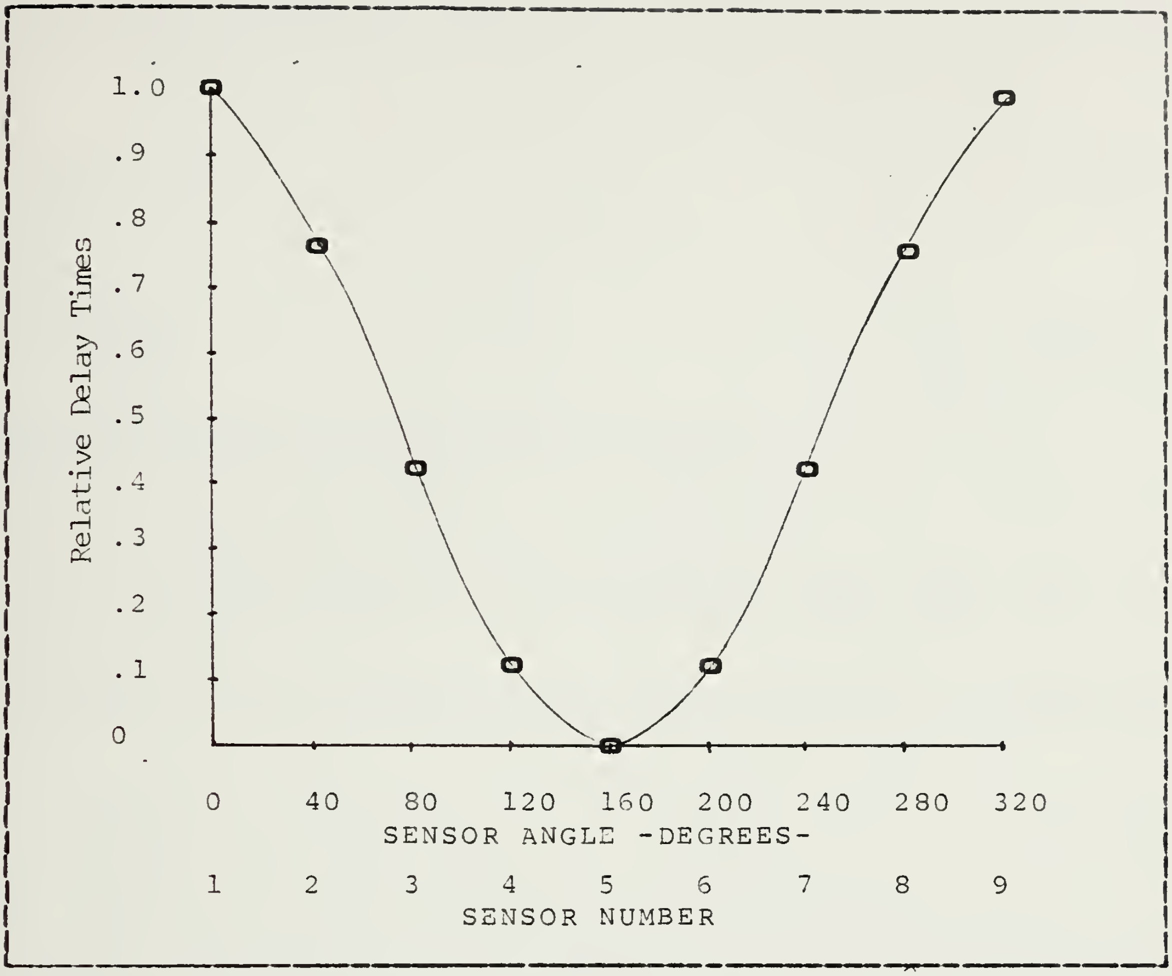

Figure 5.4 Relative Time Delay versus Sensor Angle

P_i - dependent predicted delay time values found by the least mean squares polynomial

A_i, B_i - coefficient values for the system of simultaneous equations

E_i - error between the experimental data values and predicted values of time delays

S - sum of the square errors between each data and predicted points

The derivation of the least mean squares polynomial follows:

$$S = E_1^2 + E_2^2 + E_3^2 + \dots + E_N^2$$

$$P_i = A_0 + A_1 X_i + A_2 X_i^2 + A_3 X_i^3 + \ldots + A_n X_i^n \tag{5.2}$$

$$E_i = Y_i - P_i \tag{5.3}$$

$$S = \sum_{i=1}^{N} E_i^2 \tag{5.4}$$

Combining equations 5.2 and 5.3 yields

$$E_i = Y_i - A_0 - A_1 X_i - A_2 X_i^2 - \ldots - A_n X_i^n$$

where n is the degree of the polynomial such that $N > n + 1$ and $1 < i < N$

Equation 5.4 now becomes, after substituting for E, equation 5.5

$$S = \sum_{i=1}^{N} (Y_i - A_0 - A_1 X_i - A_2 X_i^2 - \ldots - A_n X_i^n)^2 \tag{5.5}$$

To find the minimum of the sum of the squares expressed by equation 5.5, the partial derivatives of S with respect to all of the coefficients are taken. At the minimum, these partial derivatives all vanish.

$$\partial S/\partial A_0 = 0 = \sum_{i=1}^{N} 2\,(Y_i - A_0 - A_1 X_i - \ldots - A_n X_i^n)(-1)$$

$$\partial S/\partial A_1 = 0 = \sum_{i=1}^{N} 2\,(Y_i - A_0 - A_1 X_i - \ldots - A_n X_i^n)(-X_i)$$

$$\vdots$$

$$\partial S/\partial A_n = 0 = \sum 2\,(Y_i - A_0 - A_1 X_i - \ldots - A_n X_i^n)(-X_i^n)$$

Dividing by two and rearranging gives n + 1 normal simultaneous equations. Expressed in matrix notation these equations become

$$\begin{bmatrix} N & \sum X_i & \sum X_i^2 \cdots & \sum X_i \\ \sum X_i & \sum X_i^2 & \sum X_i^3 \cdots & \sum X_i \\ \sum X_i^2 & \sum X_i^3 & \sum X_i^4 \cdots & \sum X_i \\ & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot \\ \sum X_i^n & \sum X_i^{n+1} & \sum X_i^{n+2} \cdots & \sum X_i^{2n} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ \\ \\ A_n \end{bmatrix} = \begin{bmatrix} \sum Y_i \\ \sum X_i Y_i \\ \sum X_i^2 Y_i \\ \\ \\ \sum X_i^n Y_i \end{bmatrix}$$

The coefficient matrix in the above system of equations can be solved for. The minimum value for P_i can then be found. The X corresponding to this minimum value is declared to be the predicted arrival angle of the seismic wave. [Ref. 10]

4. Adaptive Target Direction Finding

An adaptive method is used to improve the directions found for all single target cases. To enhance the resolution accuracy of the peak filter output position, the single target seismic data is copied into the filter data section for its class of target. Subsequently, when the matched filter routine is called for each sensor, filtering is performed with sensor data from that time period.

5. Software features of the Multiple Target Direction Routine

As first addressed in the previous chapter, the multiple direction routine allows for the selection of the number of data points to be used from the 1024 size buffer of sensor data. This was included to investigate the algorithm's performance for various data/filter window sizes and to allow an option for reduced CPU time utilzation for interactive program runs. Tabular output from this routine includes the event number, the time period for the data, identification and directions for up to five target classes, and up to four simulated target specifications. Also, notice that the routine adapts to the number of sensors

specified for the ring. This was necessary since the experimental data included sensor rings of six and nine vertical sensors.

C. MULTIPLE TARGETS OF THE SAME TARGET CLASS

A limitation on the system, as presented up to this point, is its inablity to engage multiple targets of the same class. Discrete targets may appear as separate entities since the seismic signals generated by the two same class targets are not likely to be incident at the same time. This situation is greatly complicated for continuous time signals, such as tracked or wheeled vehicles. For such continuous time targets, mutual distortion would be the likely result.

An algorithm is needed to identify these multiple peaks without erroneously declaring elements in the same peak as targets. By using only the positive half of the matched filter output and performing data smoothing on the remaining points, a curve with its number of peaks equalling the number of targets present could be generated. Numerical methods for curve fitting or interpolation are available [Ref. 11.]. Polynomial curve fitting, Sterling's method and variations of Newton's method are only a few of the possible approaches applicable for equally spaced data. By differentiating the smoothed data, the peaks and valleys of the filter output can be found. The height of the curve corresponding to the points where the derivative is zero can now be compared to the selected matched filter threshold. This excludes the valley points and leaves the points remaining which correspond to the relative times of the same class targets.

This method is constrained by resolution of close proximity, time limited targets and near phase synchronization

of continuous targets. The variables to be optimized through experimentation may be the degree of the smoothing of the curve data and the exclusion of erroneous valleys associated with the same target's data.

VI. ANALYSIS OF SEISMIC DATA

Analysis of the simulated and experimental seismic data will be conducted as detailed in Table II. Table III lists the matched filter contents for the simulated data. Table IV is the test plan for the experimental data. Table V lists the matched filter signals used for the experimental data analysis.

Table VI summarizes the results of the simulted and experimental data runs for direction finding. The window size, used for all multiple target direction finding results, was 300. Table VII lists the events in which targets were missed or incorrectly identified.

The time domain phase difference directions found, are not presented for the reasons noted earlier. Errors of up to eighty degrees were not uncommon with this method.

Each event run will be accompanied by the following graphic output:

1. Least Mean Square Initial Direction
2. Matched Filter Response
3. Amplitude Response
4. Amplitude Response of any Malfunctioning Sensor
5. Frequency Response
6. Least Mean Squares Polynomial Curve Fitting (LMSP) Using Matched Filter Outputs
7. Multiple Target Direction Summary Resulting from Least Mean Squares Curve Fitting

TABLE II

Test Plan for Simulated Data

Event	#Sen	#Tgts	Frequency	Amplitude	Direction
001	9	1	10	3000	0
001	9	1	10	3000	40
001	9	1	10	3000	120
001	9	1	10	3000	240

TABLE III

Matched Filter for Simulated Targets

Filter	Frequency	Amplitude	Direction
Tracked Veh	30	2000	0
Wheeled Veh	10	2000	0
Shell blast data from event #383			0
Helicopter	15	2000	0
Personnel	20	2000	0

TABLE IV

Test Plan for Experimental Data

Event	#Sen	#Tgts	Dir	Target	Distance
383	9	1	0	Shot	5KM
382	9	1	0	Shot	5Km
372	6	1	315	Helicopter	5 - 15KM
375	6	1	0	Tank	5 - 0Km
374	6	1	315	Helicopter	15 - 5KM
302	6	1	0	Mortar	1KM
314	6	1	315	LVT	4 - 5KM
354	6	5	0 225 315 0	105mm How 175mm Gun LVT M-60 Tank	5Km 4KM 4 - 5KM 4 - 5KM

TABLE V

Matched Filter for Experimental Data

Target	Event Used as Filter
Tracked Vehicle	375
Wheeled Vehicle	none (backround noise)
Blast/Recoil	383
Helicopter	372
Personnel	none (backround noise)

TABLE VI

Summary of Direction Finding Results

Event	#Sen	#Tgts	Distance	Dir	Initial Dir	∠error	LMSP Dir	∠error
001	9	1	N/A	0	N/A		0	0
001	9	1	N/A	40	N/A		40	0
001	9	1	N/A	120	N/A		120	0
001	9	1	N/A	240	N/A		240	0
383	9	1	5KM	0	4(3)	1.1	28(4) -5	7.8 1.4
382	9	1	5KM	0	-14(4)	3.9	-6	1.67
372	6	1	5 - 15KM	315	-32	3.6	-59	3.9
375	6	1	5 - 0KM	0	312	13.3	FAILED	
374	6	1	15 - 5KM	315	-59	4.0	FAILED	
302	6	1	1KM	0	4	1.1	6	1.7
354	6	5	5KM 4 KM 4 - 5KM 4 - 5KM	0 225 315 0	-3 291 FAILED FAILED	.8 18.0	3 FAILED	.8

*Note: The matched filter threshold was set at .9 for all single targets and .6 for all multiple target events. Brackets indicate the use of other than a second degree polynomial.

TABLE VII

Missed or Incorrectly Identified Targets

Event	#Targets	Target	Nature of Errors
375	1	Tank	1. Matched filter was not based on a high S/N sample signal
374	1	Helo	2. Small seismic signal amplitudes
354	1	LVT	3. Malfunctioning sensor(s)
	1	Tank	
	1	175mm Gun	Sever clipping distortion of input signal

Note: The numbered error sources apply to all events listed.

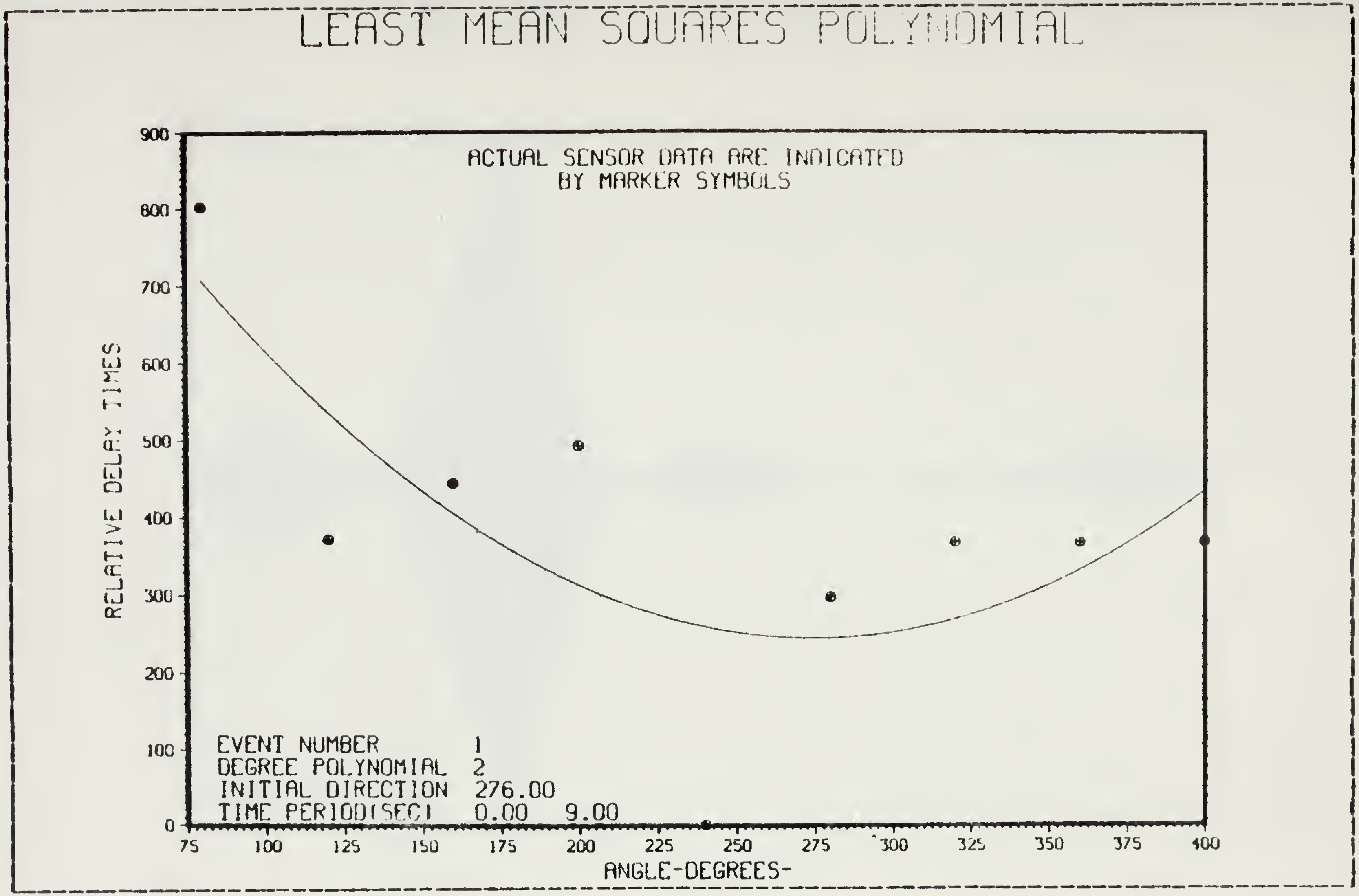

Figure 6.1 Sample Least Mean Squares Initial Direction for Event 001

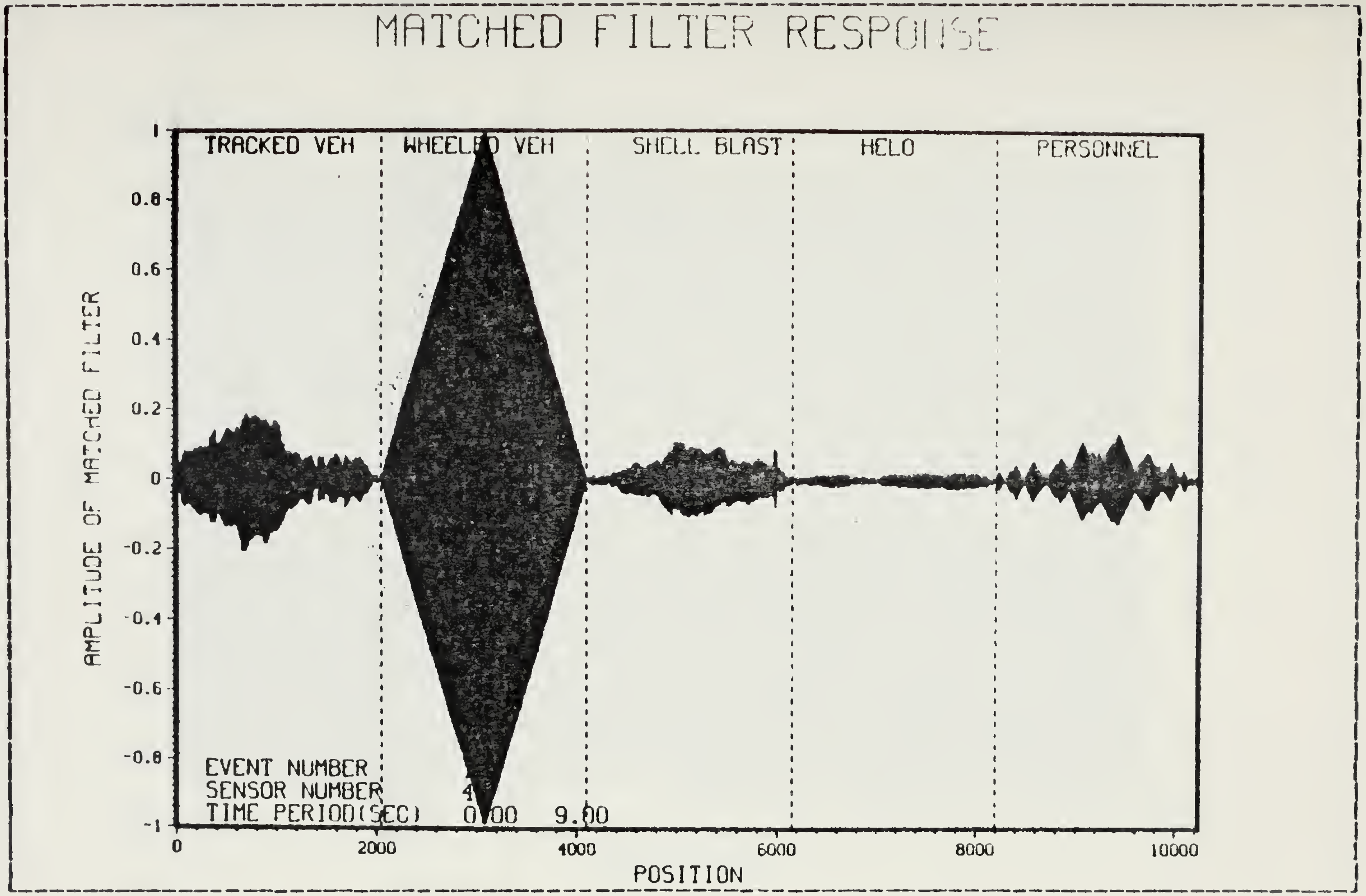

Figure 6.2 Sample Matched Filter Response for Event 001

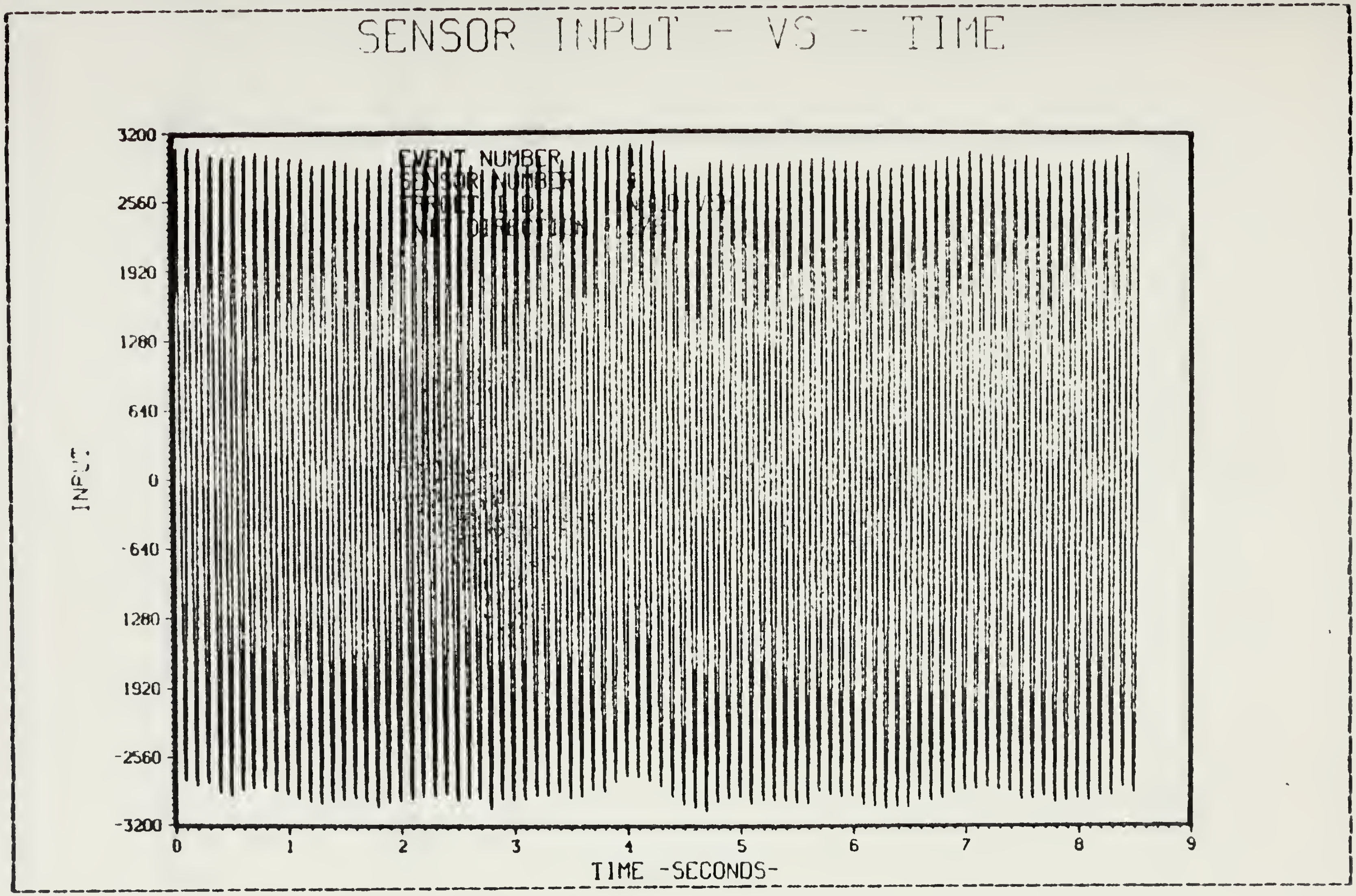

Figure 6.3 Sample Amplitude Response for Event 001

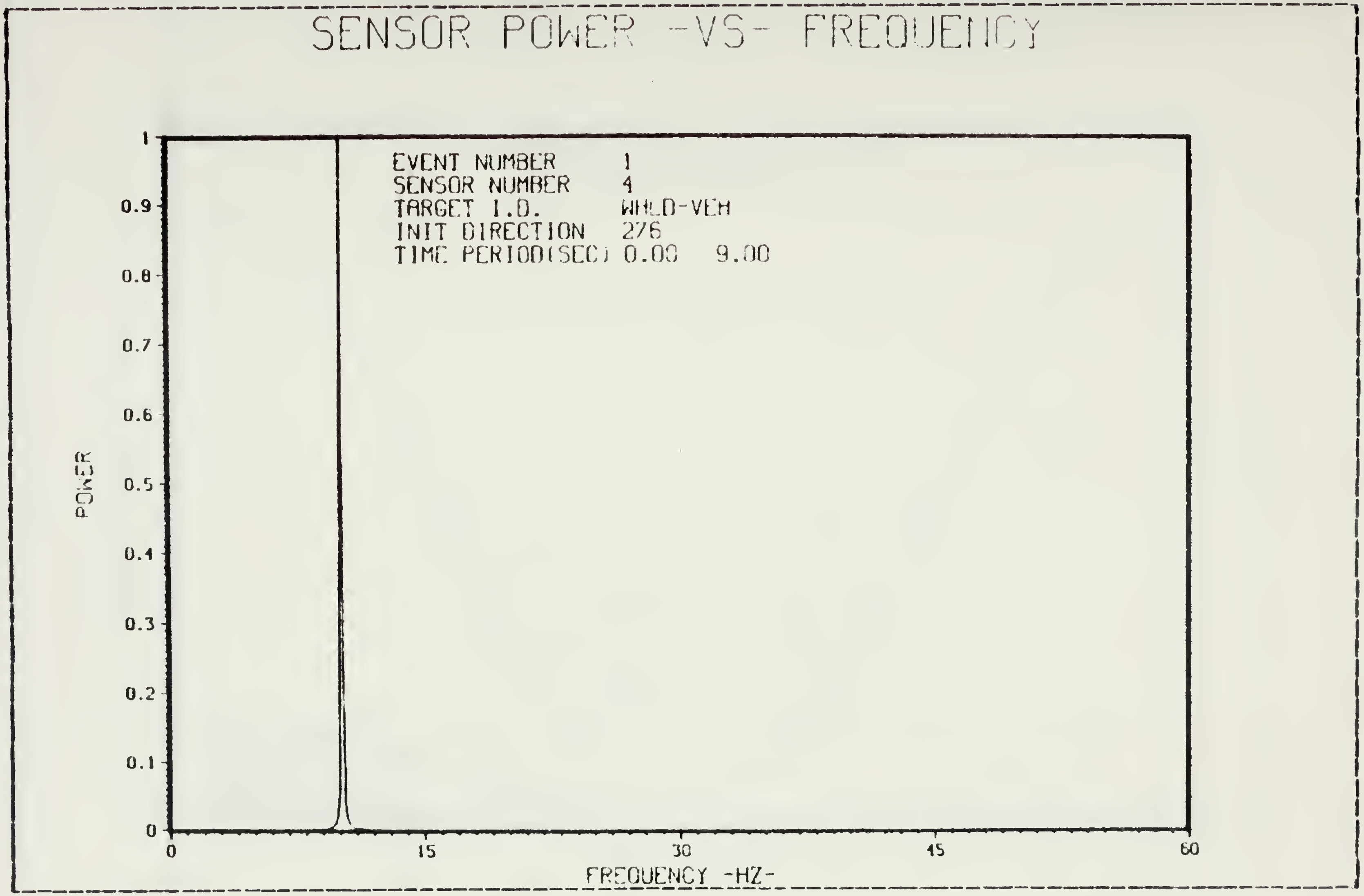

Figure 6.4 Sample Frequency Response for Event 001

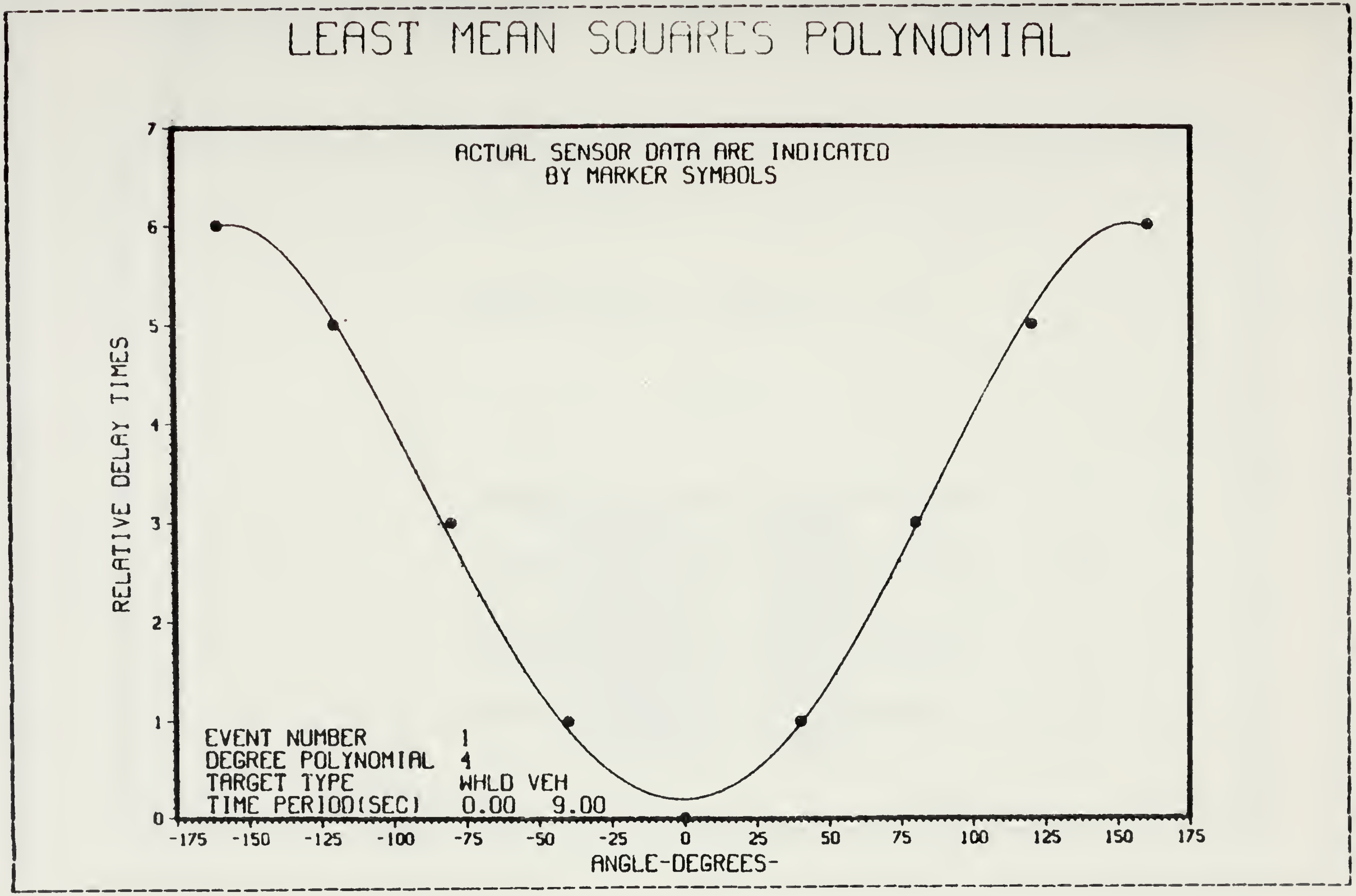

Figure 6.5 LMSP Matched Filter Direction for Event 001

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```
EVENT NUMBER     1

TIME PERIOD(SEC) 0.00   9.00


WHEELED VEHICLE   DIRECTION - 0.00


SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    10.00
        AMPLITUDE     3000.0000
        DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 6.6   LMSP Multiple Target Direction Summary for Event 001

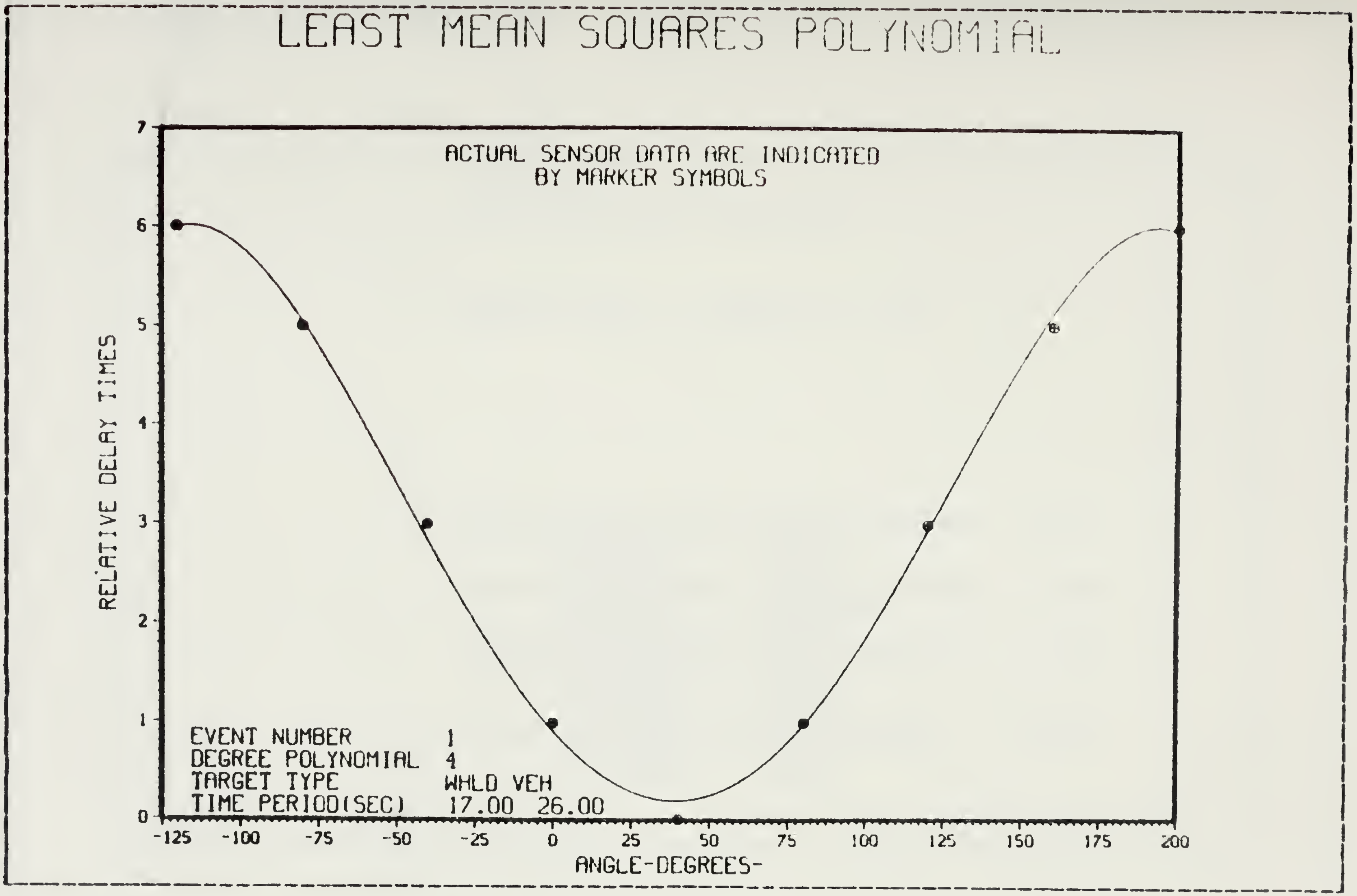

Figure 6.7 LMSP Matched Filter Direction for Event 001

```
MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER      1

TIME PERIOD(SEC) 17.00  26.00


WHEELED VEHICLE   DIRECTION - 40.00




SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    10.00
        AMPLITUDE     3000.0000
        DIRECTION     40.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
        AMPLITUDE     0.0000
        DIRECTION     40.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 6.8 LMSP Multiple Target Direction Summary for Event 001

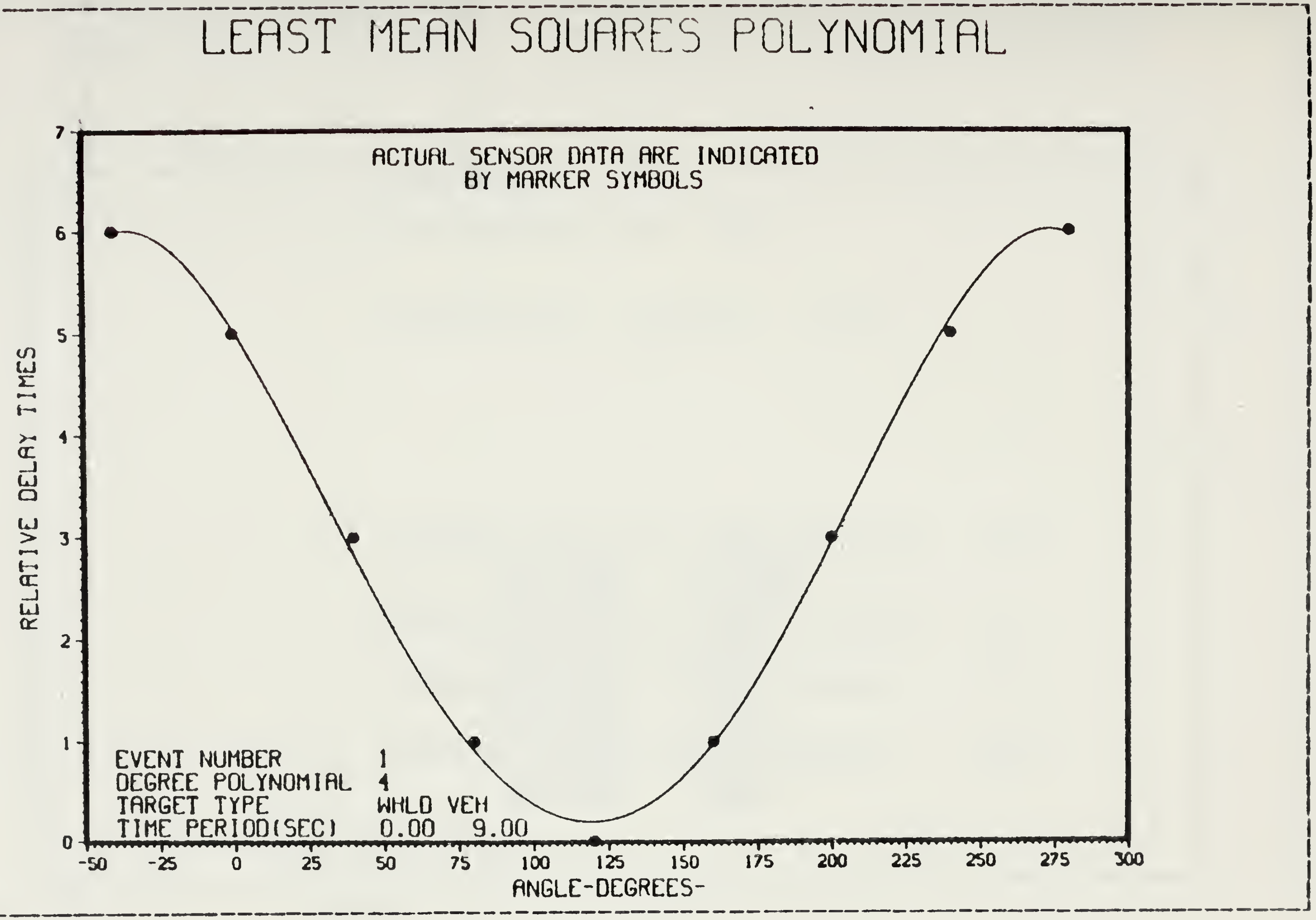

Figure 6.9 LMSP Matched Filter Direction for Event 001

```
MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER      1

TIME PERIOD(SEC) 0.00   9.00


WHEELED VEHICLE   DIRECTION - 120.00



SIMULATED TRKD VEHICLE TARGET FREQUENCY    10.00
        AMPLITUDE     3000.0000
        DIRECTION     120.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 6.10 LMSP Multiple Target Direction Summary for Event 001

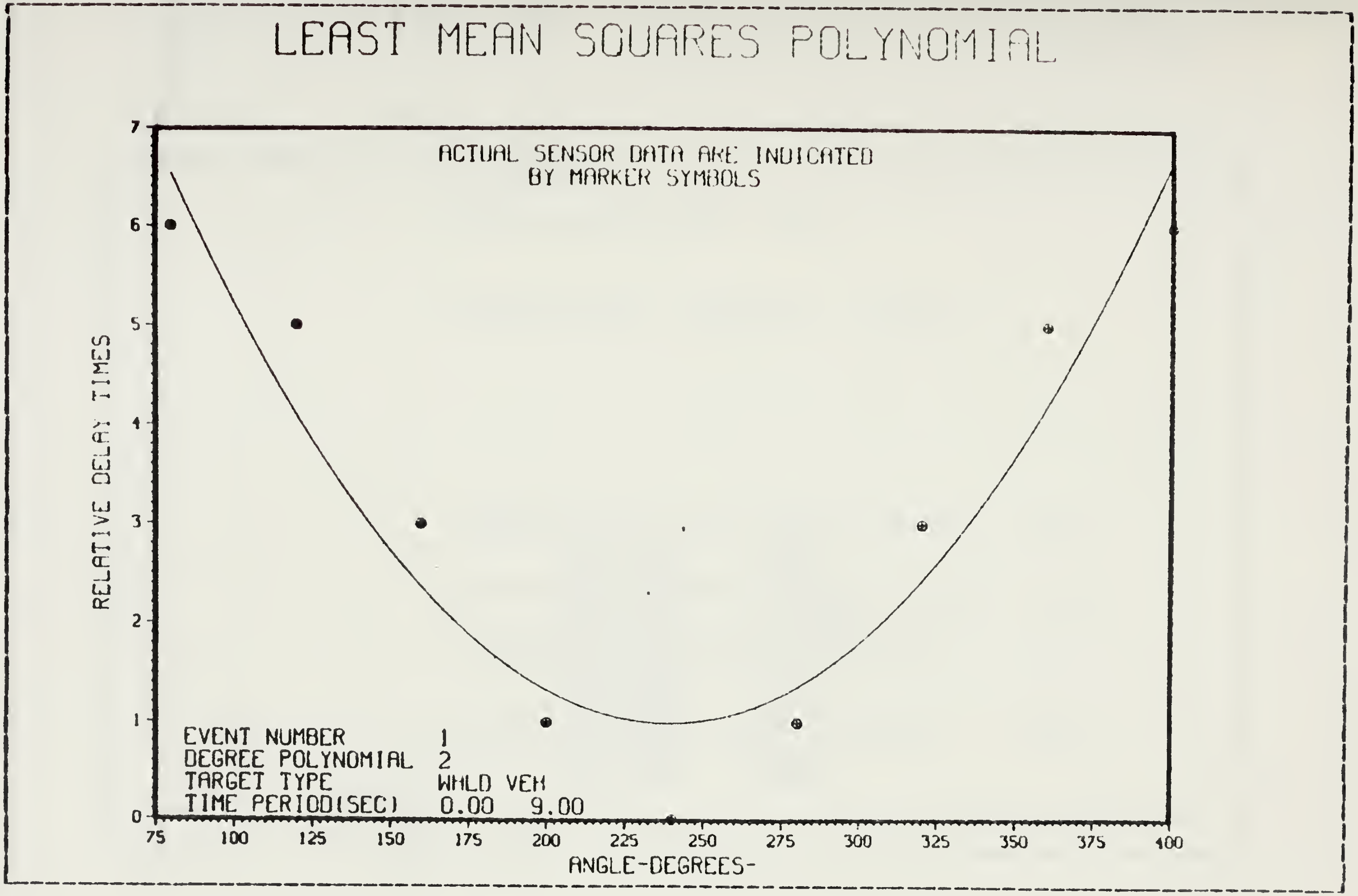

Figure 6.11 LMSP Matched Filter Direction for Event 001

```
MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER      1

TIME PERIOD(SEC) 0.00   9.00


WHEELED VEHICLE   DIRECTION - 240.00




SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    10.00
          AMPLITUDE     3000.0000
          DIRECTION     240.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
```

Figure 6.12 LMSP Multiple Target Direction Summary for Event 001

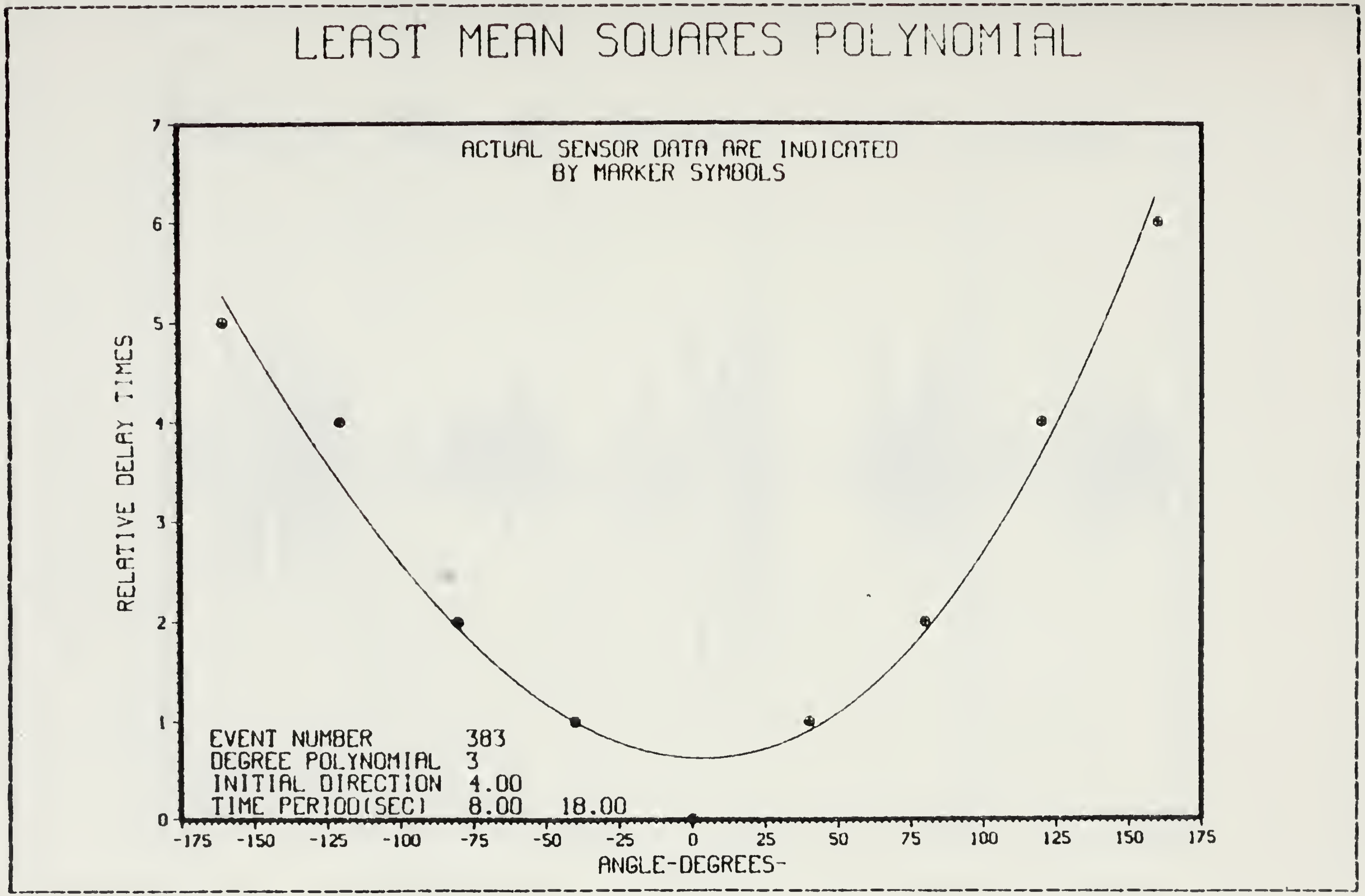

Figure 6.13 LMSP Initial Direction for Event 383

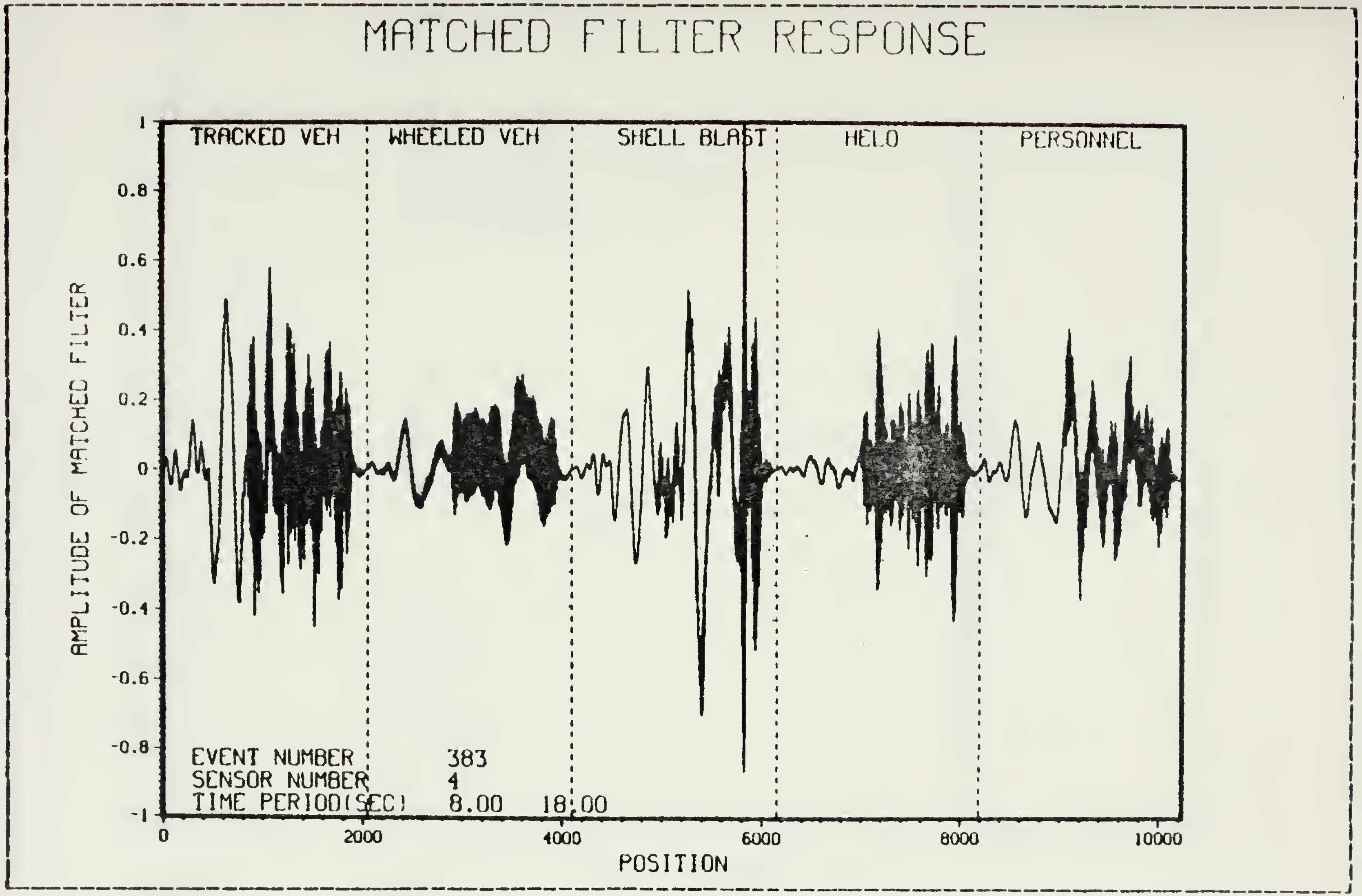

Figure 6.14 Matched Filter Response for Event 383

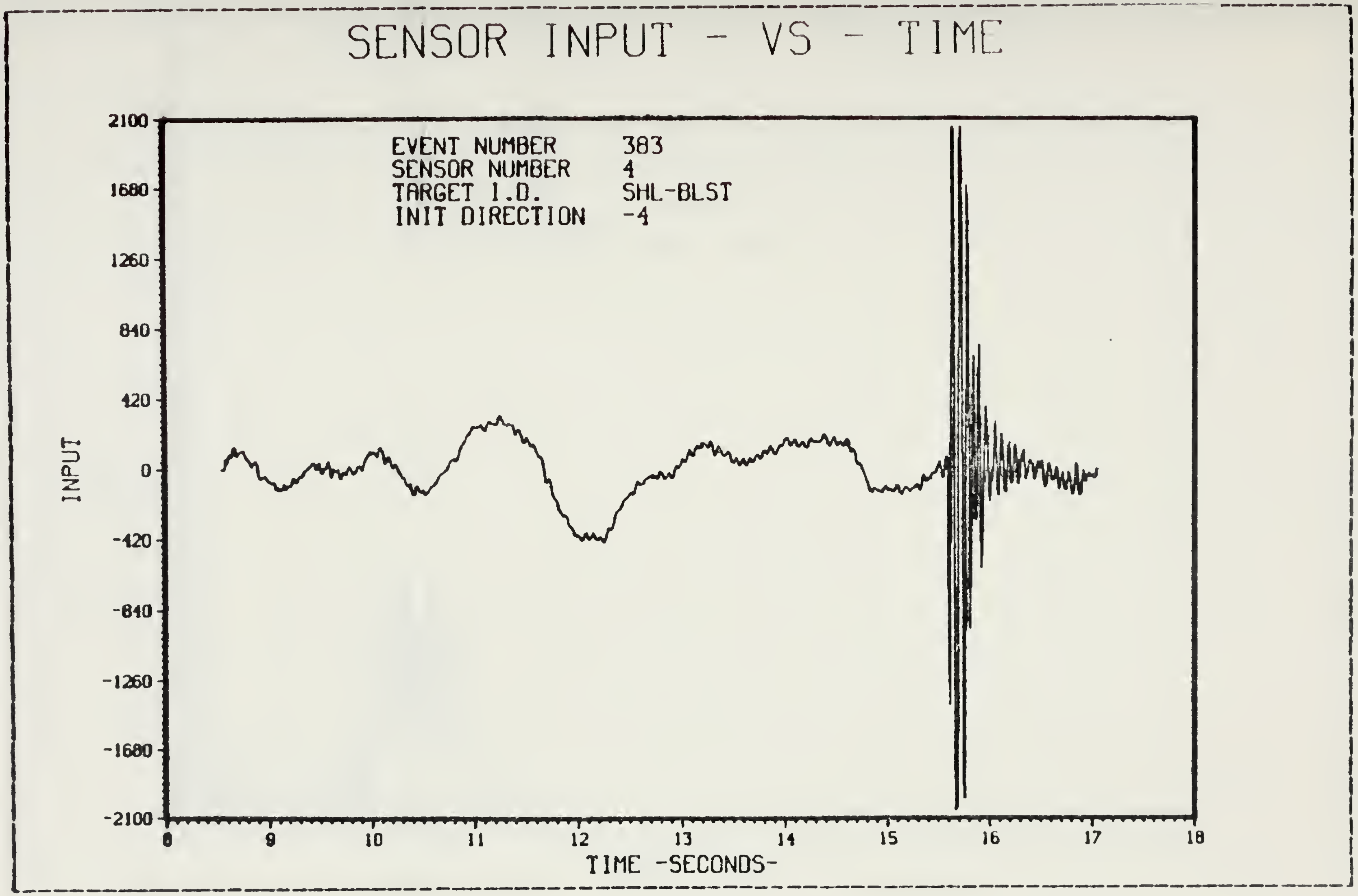

Figure 6.15 Amplitude Response for Event 383

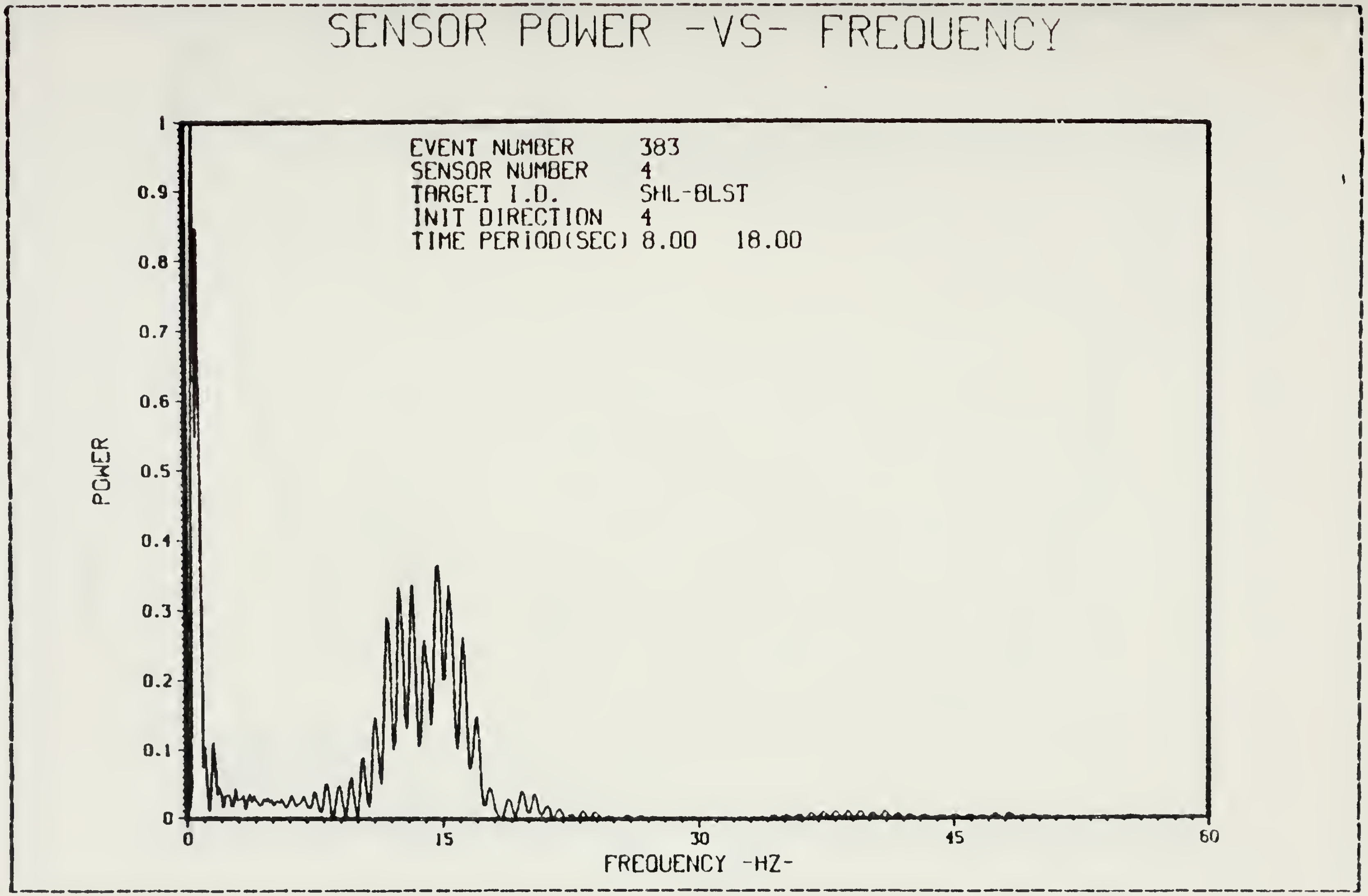

Figure 6.16 Frequency Response for Event 383

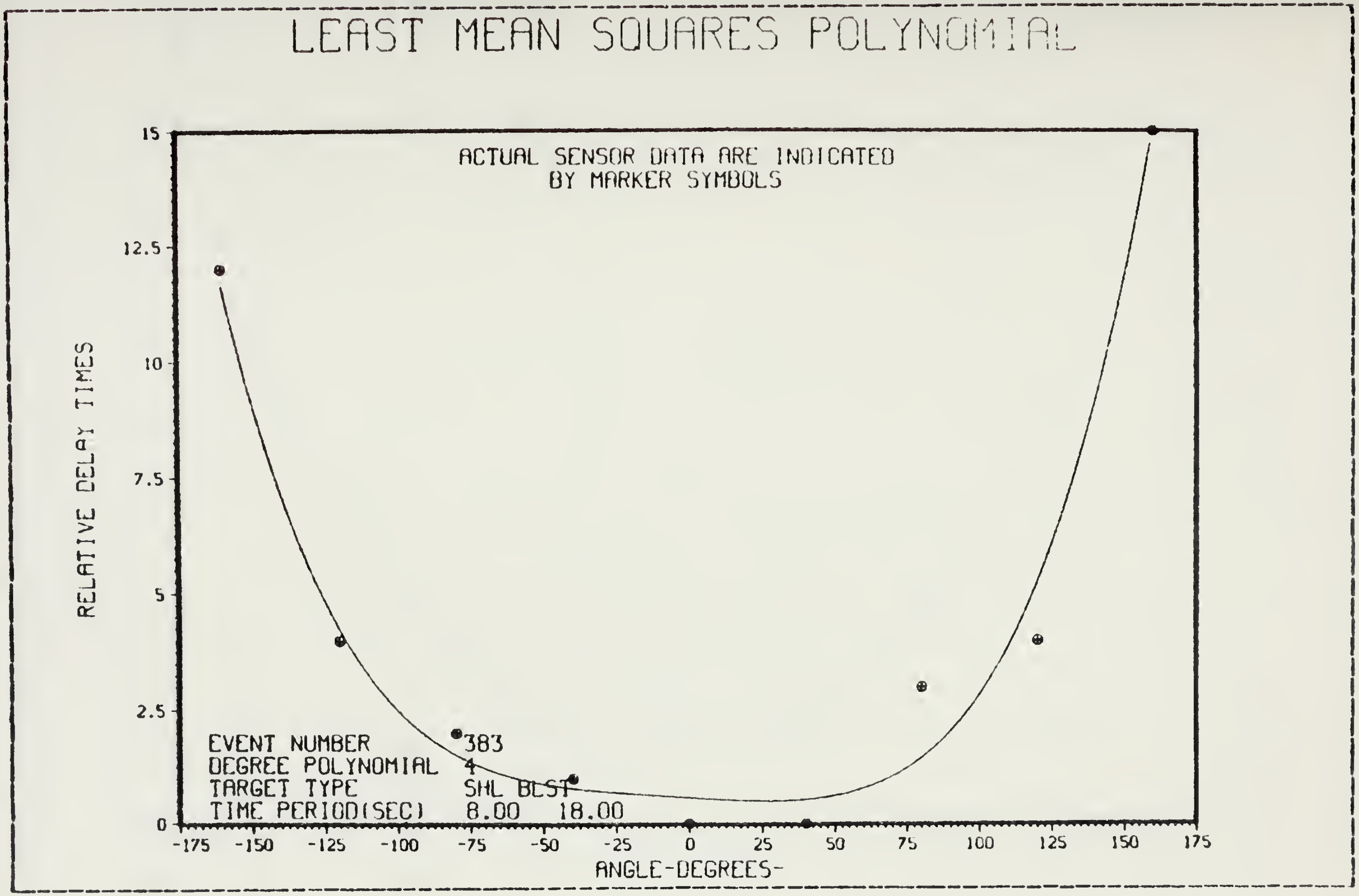

Figure 6.17 Fourth Degree LMSP Matched Filter Direction for Event 383

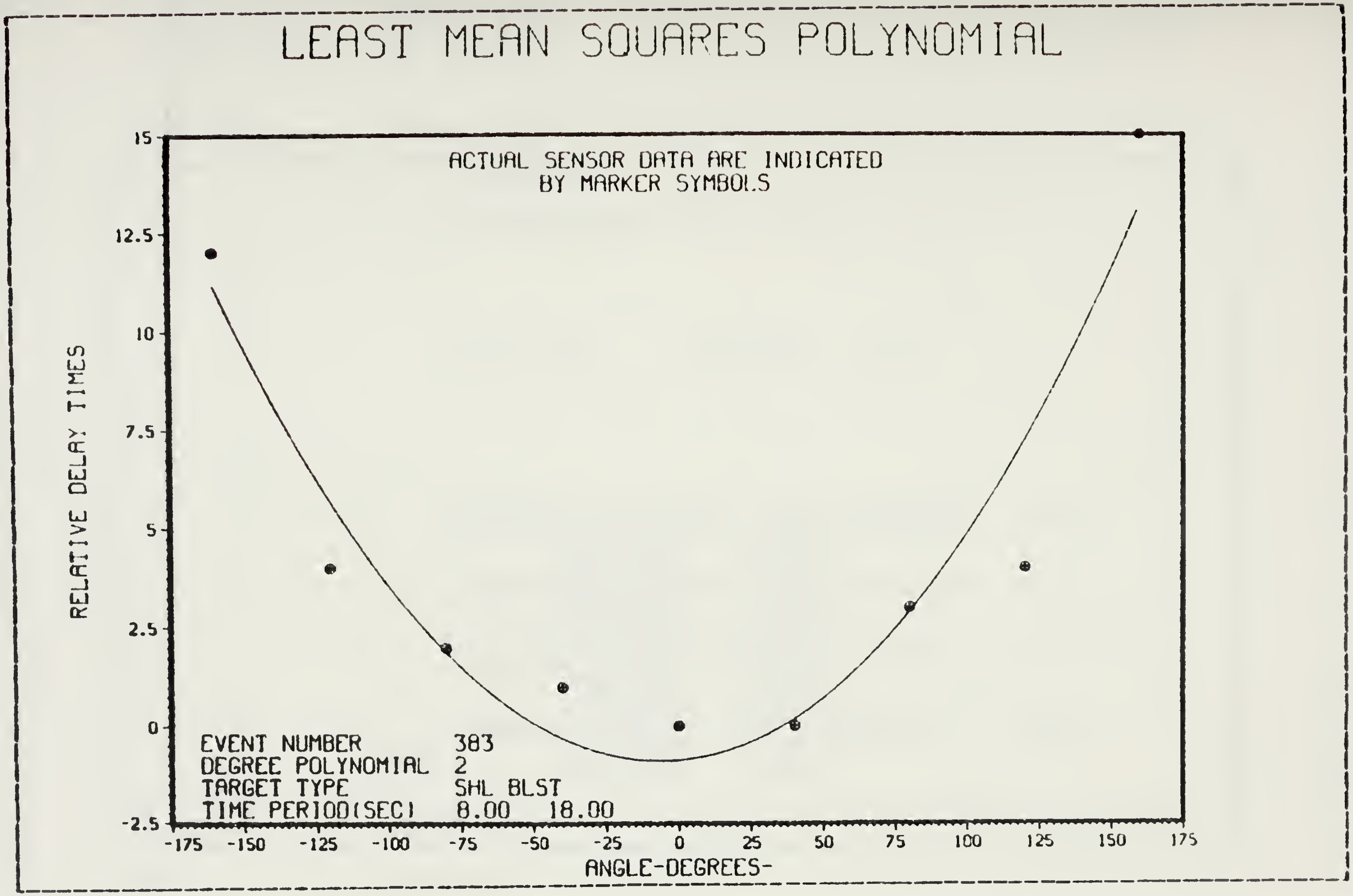

Figure 6.18 Second Degree LMSP Matched Filter Direction for Event 383

```
MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER     383

TIME PERIOD(SEC) 8.00   18.00

SHELL BLAST        DIRECTION - 28.00

SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 6.19 LMSP Multiple Target Direction Summary for Event 383

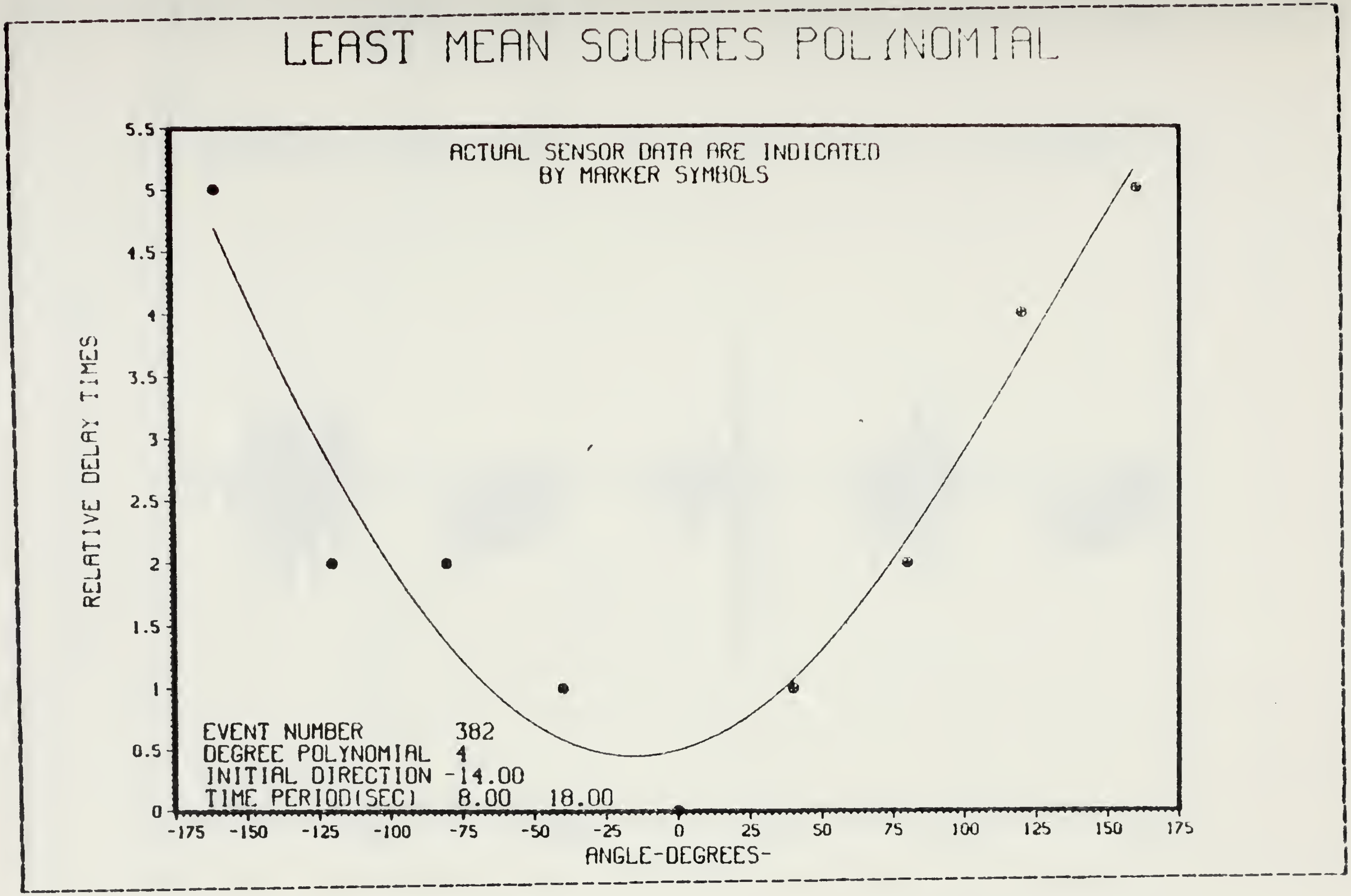

Figure 6.20 LMSP Initial Direction for Event 382

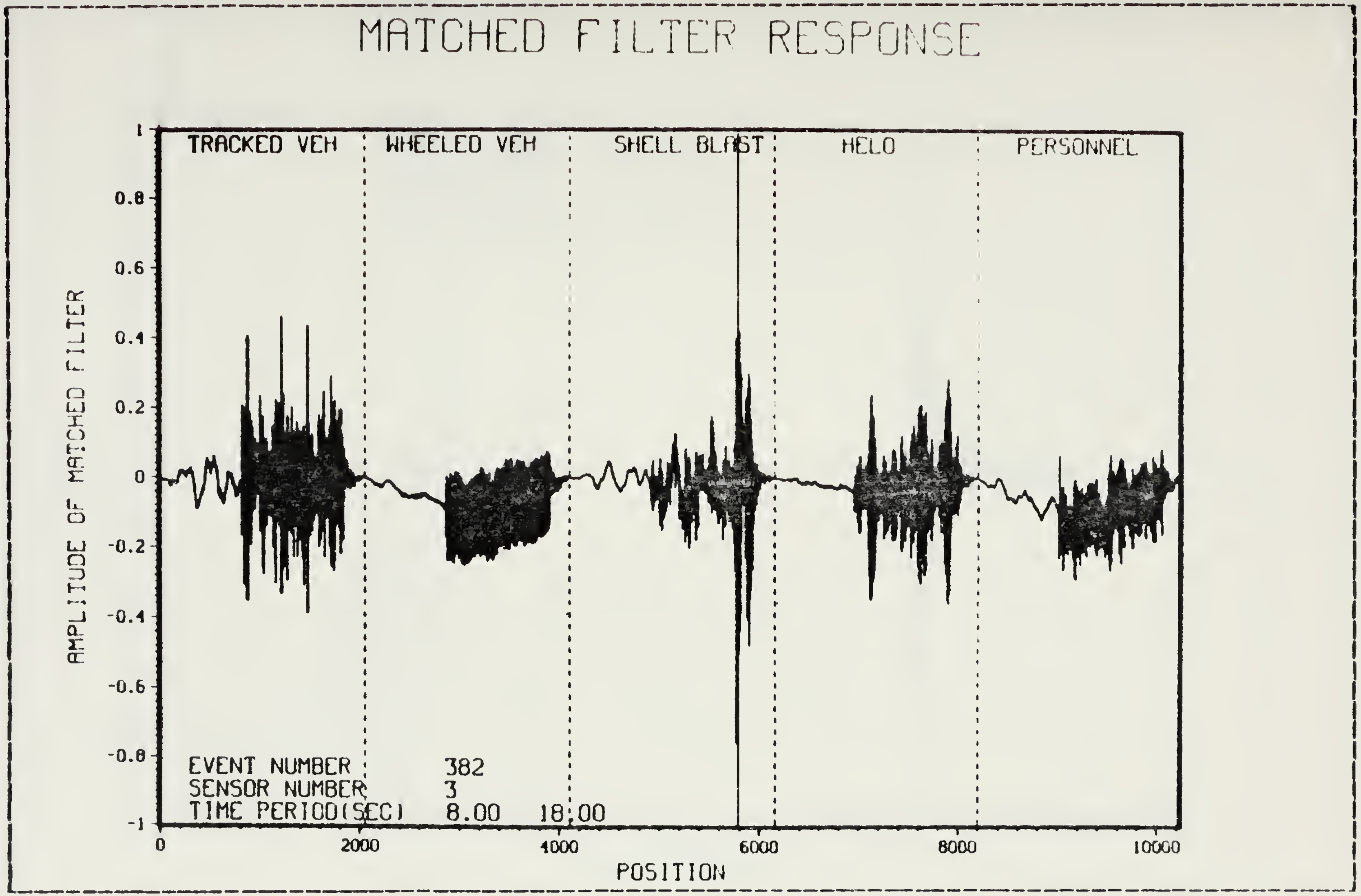

Figure 6.21 Matched Filter Response for Event 382

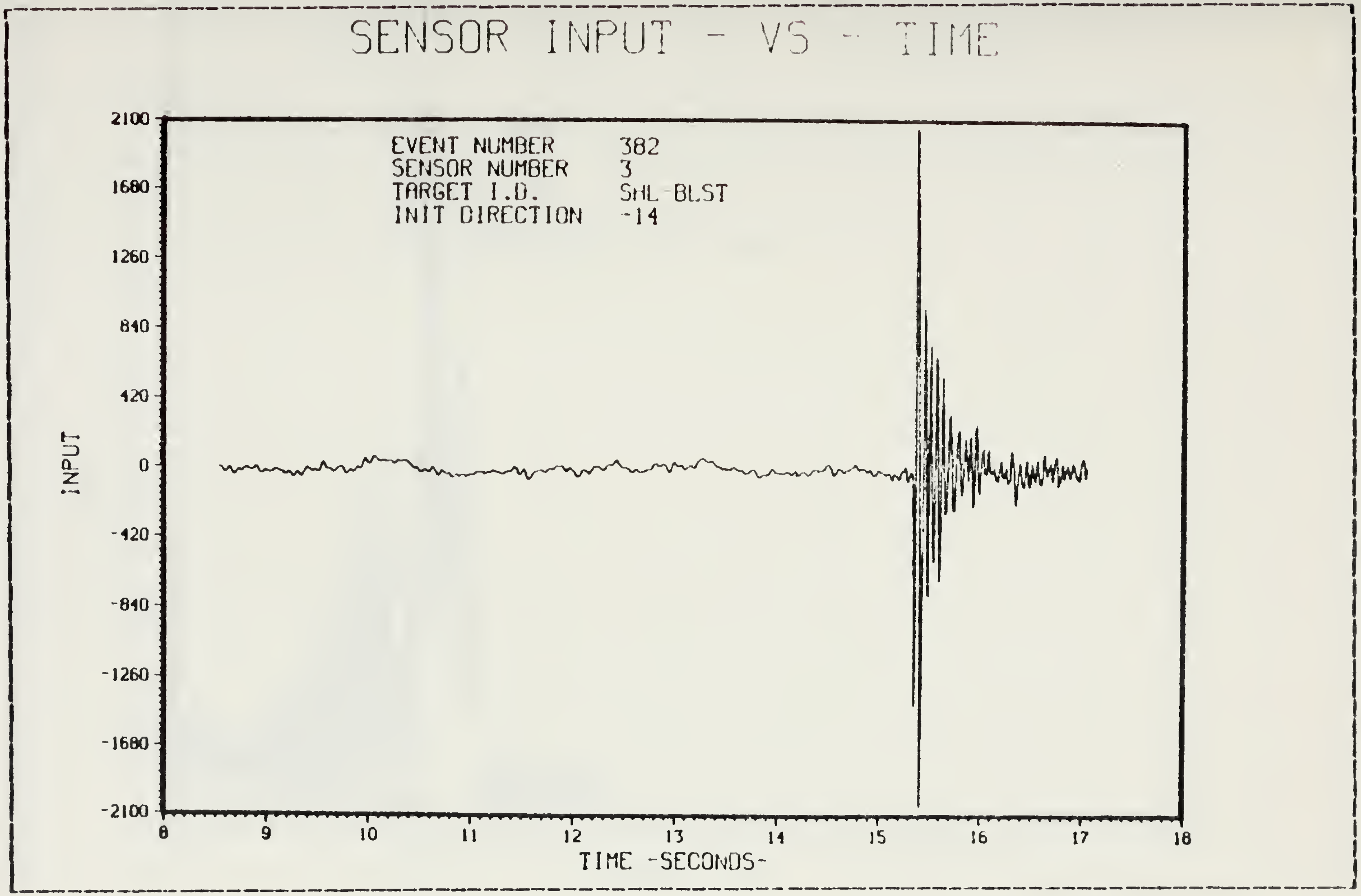

Figure 6.22 Amplitude Response for Event 382

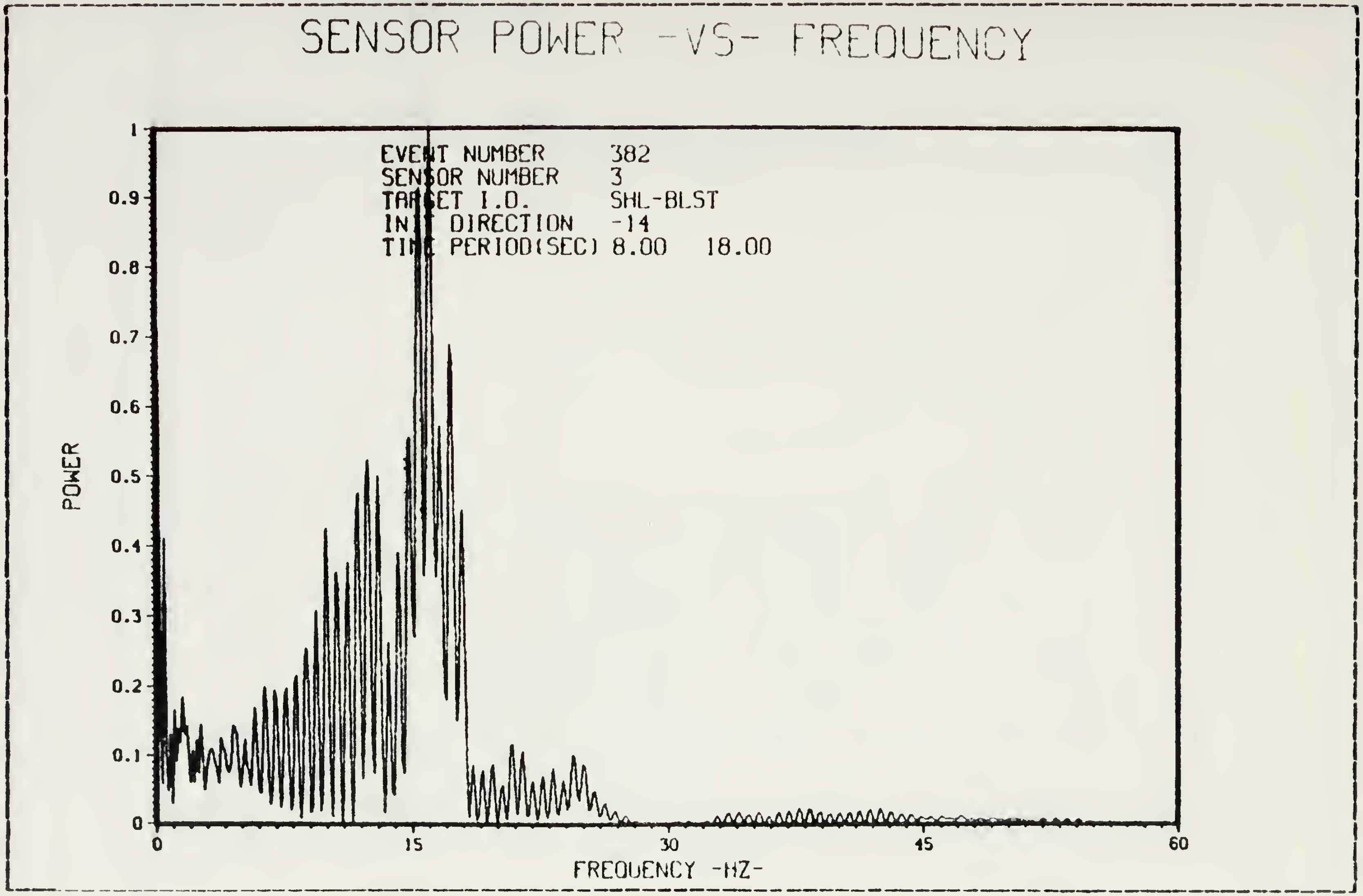

Figure 6.23 Frequency Response for Event 382

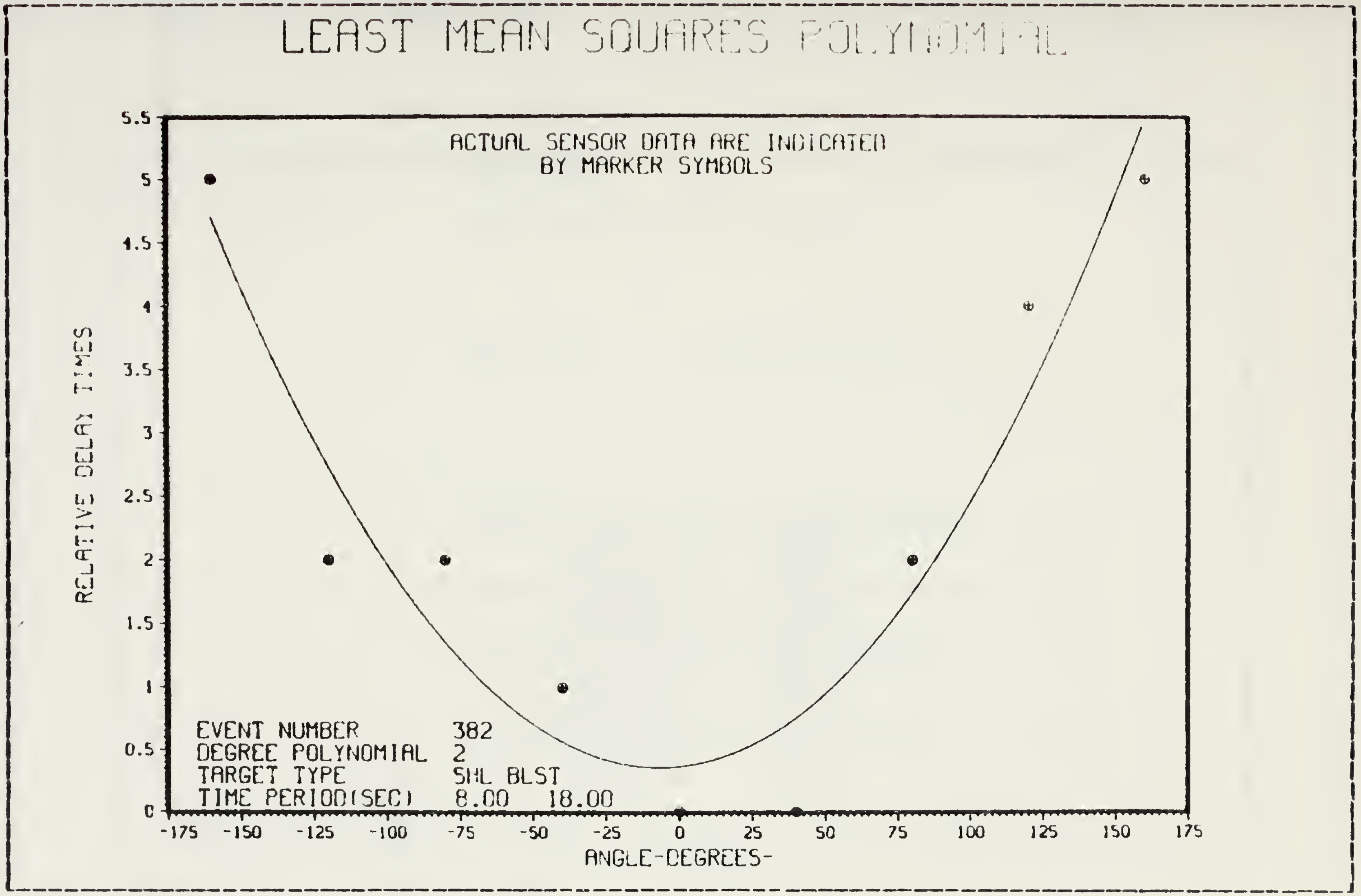

Figure 6.24 LMSP Matched Filter Direction for Event 382

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```
EVENT NUMBER     382

TIME PERIOD(SEC) 8.00   18.00



SHELL BLAST        DIRECTION - -6.00



SIMULATED TRKD VEHICLE TARGET FREQUENCY     0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY     0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY       0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY        0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
```

Figure 6.25   LMSP Multiple Target Direction Summary for Event 382

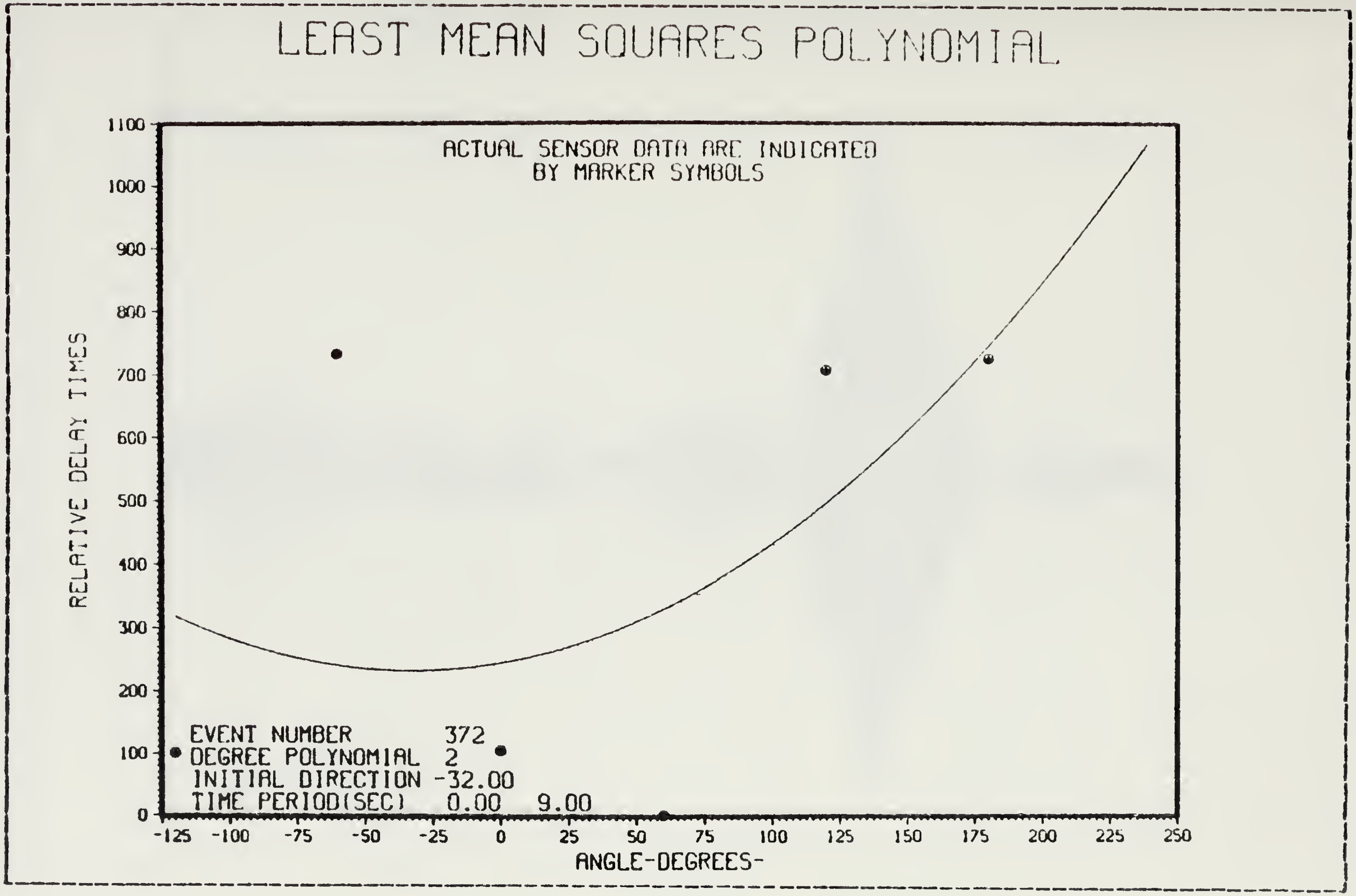

Figure 6.26 LMSP Initial Direction for Event 372

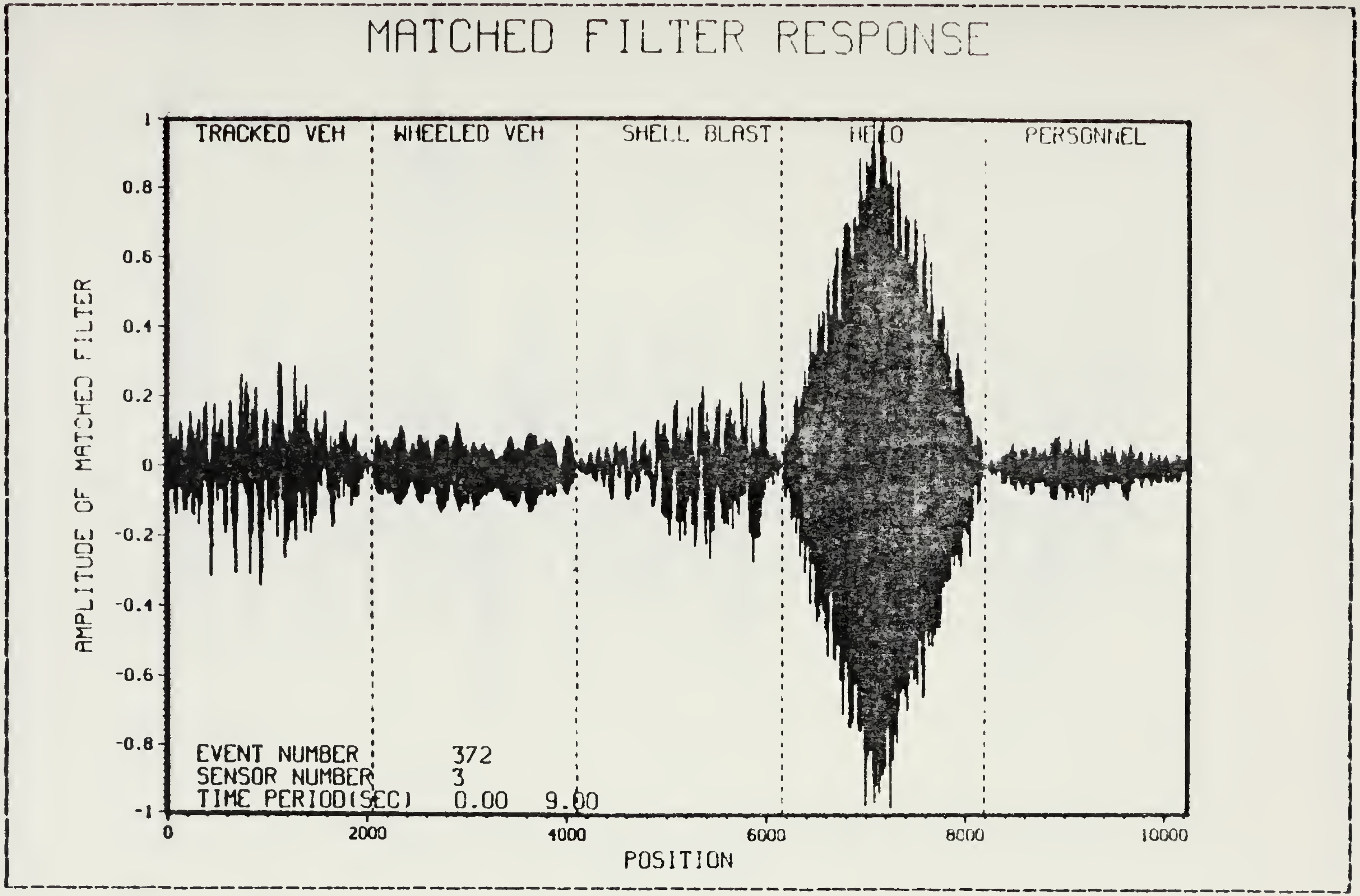

Figure 6.27 Matched Filter Response for Event 372

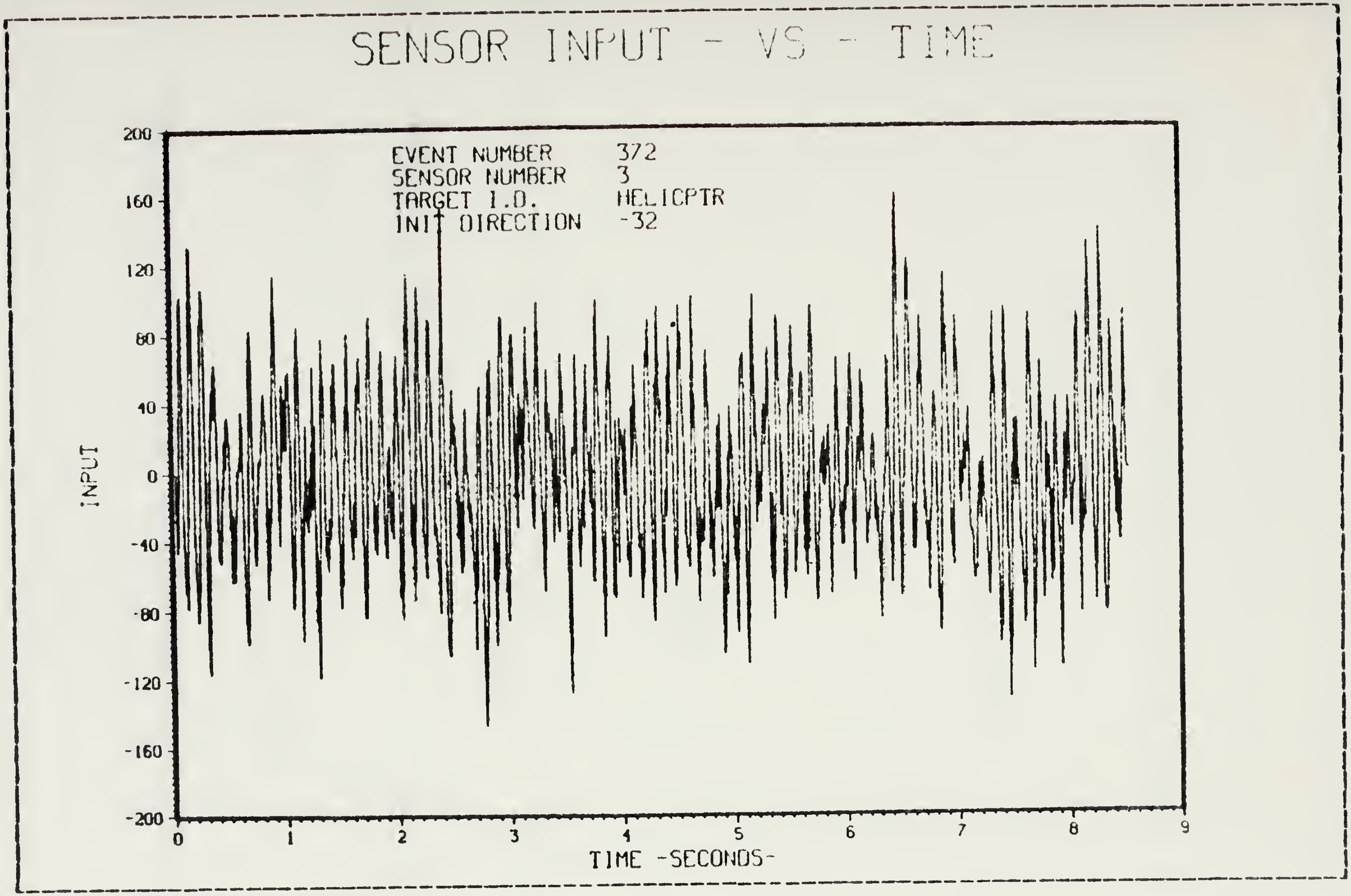

Figure 6.28 Amplitude Response for Event 372

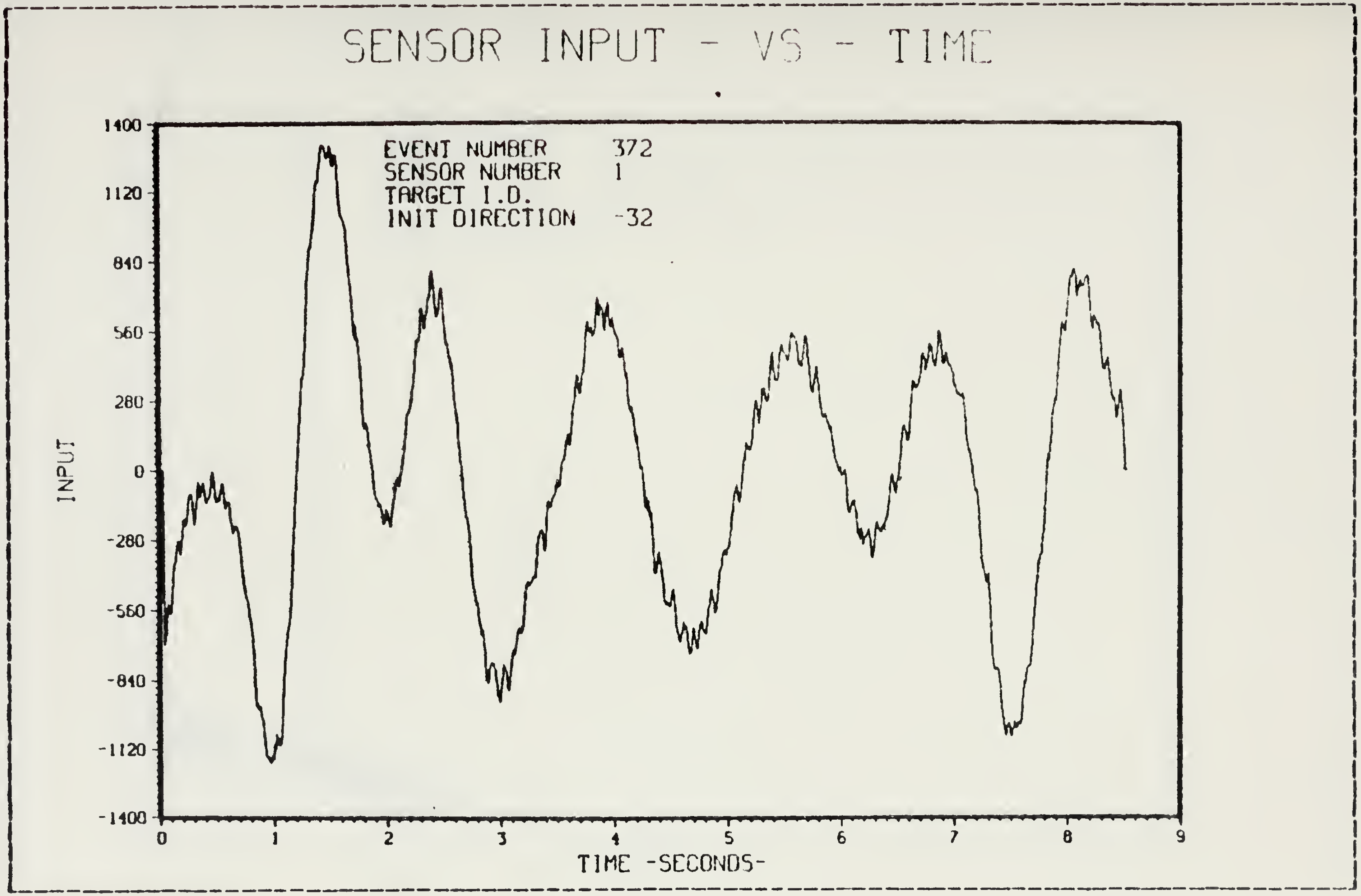

Figure 6.29 Amplitude Response of Malfunctioning Sensor for Event 372

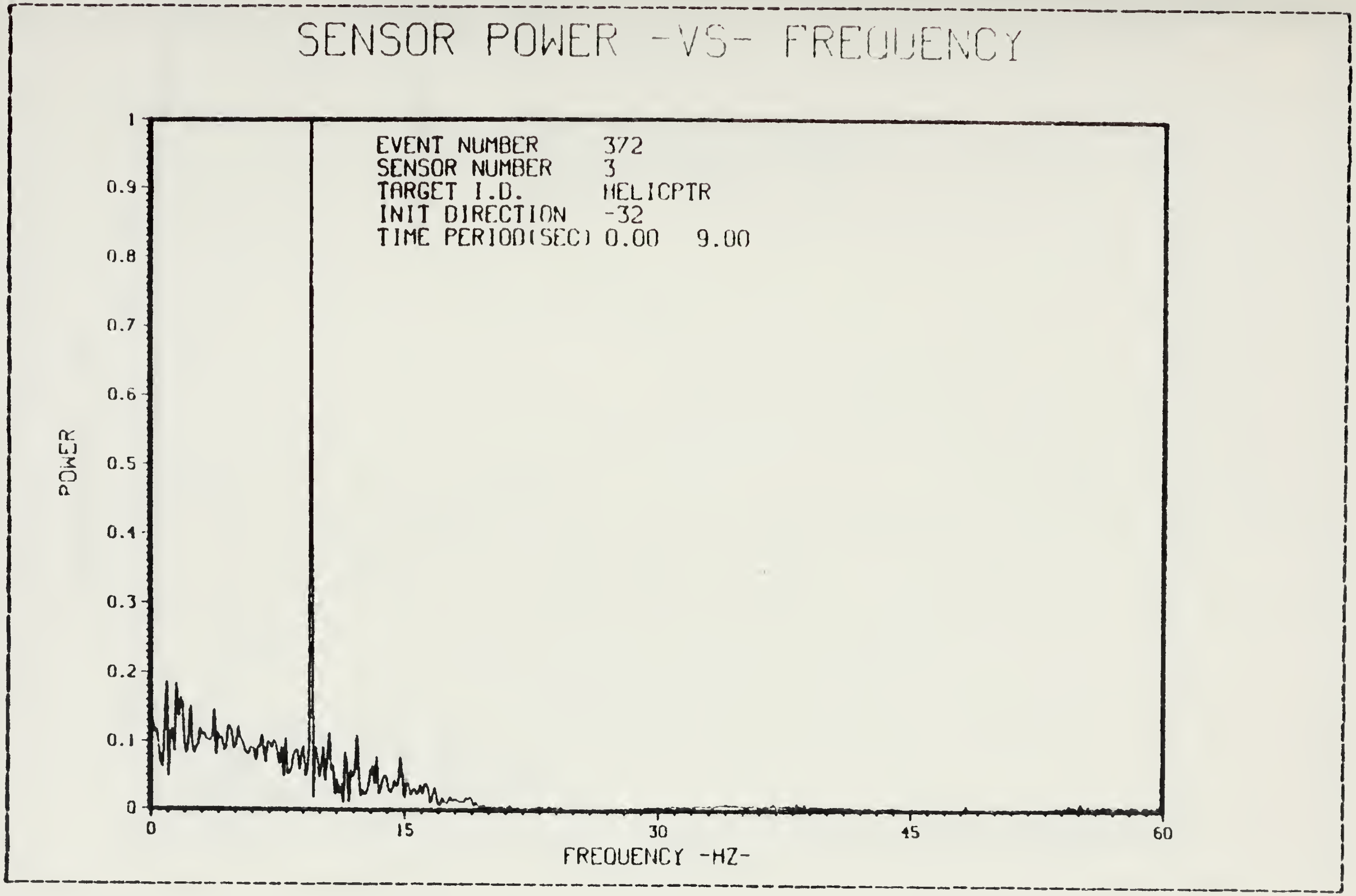

Figure 6.30 Frequency Response for Event 372

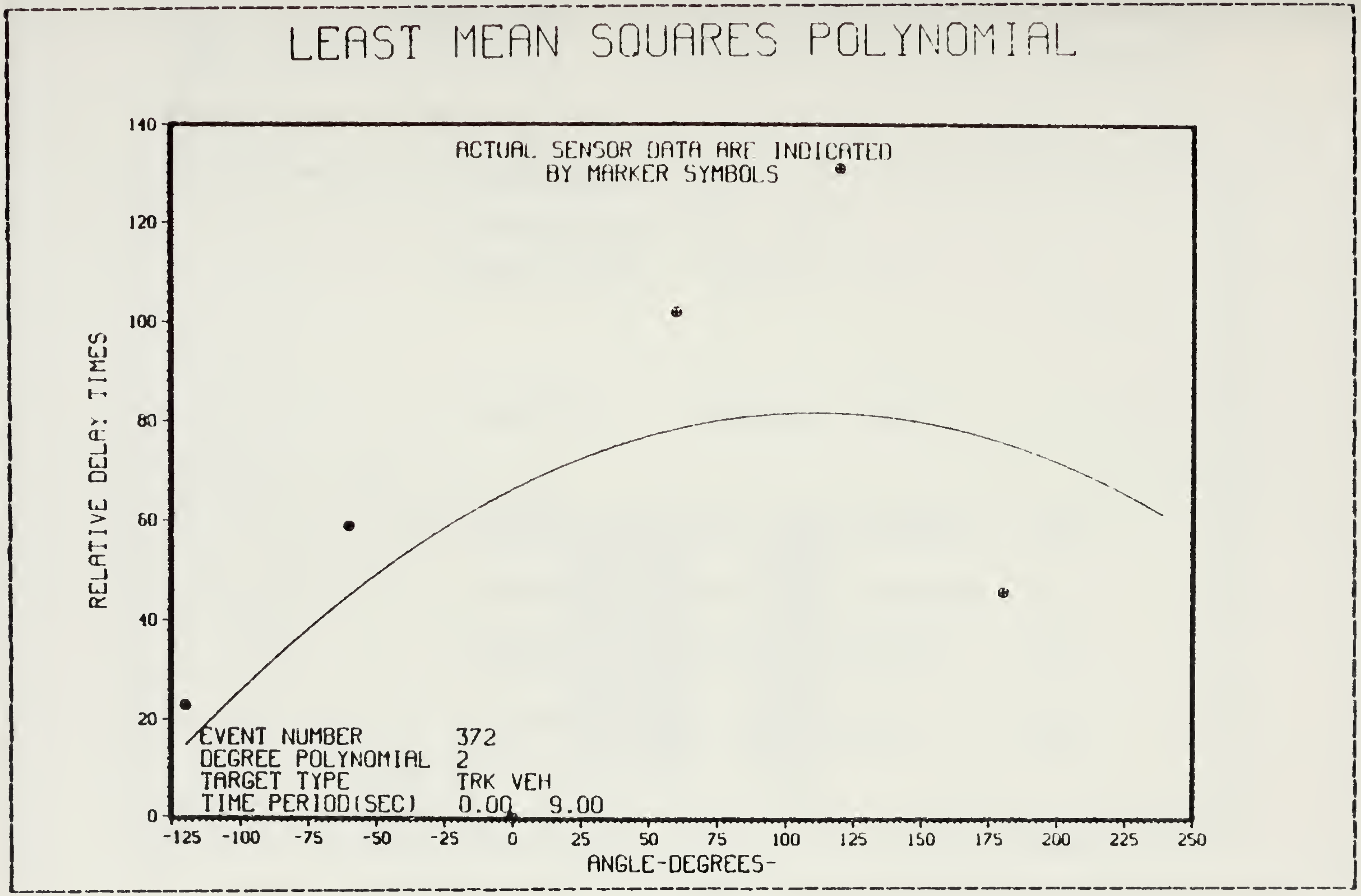

Figure 6.31 LMSP Matched Filter Direction for Event 372

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```
EVENT NUMBER     372

TIME PERIOD(SEC) 0.00   9.00

TRACKED VEHICLE   DIRECTION - 0.00



HELICOPTER        DIRECTION - 0.00



SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 6.32 LMSP Multiple Target Direction Summary for Event 372

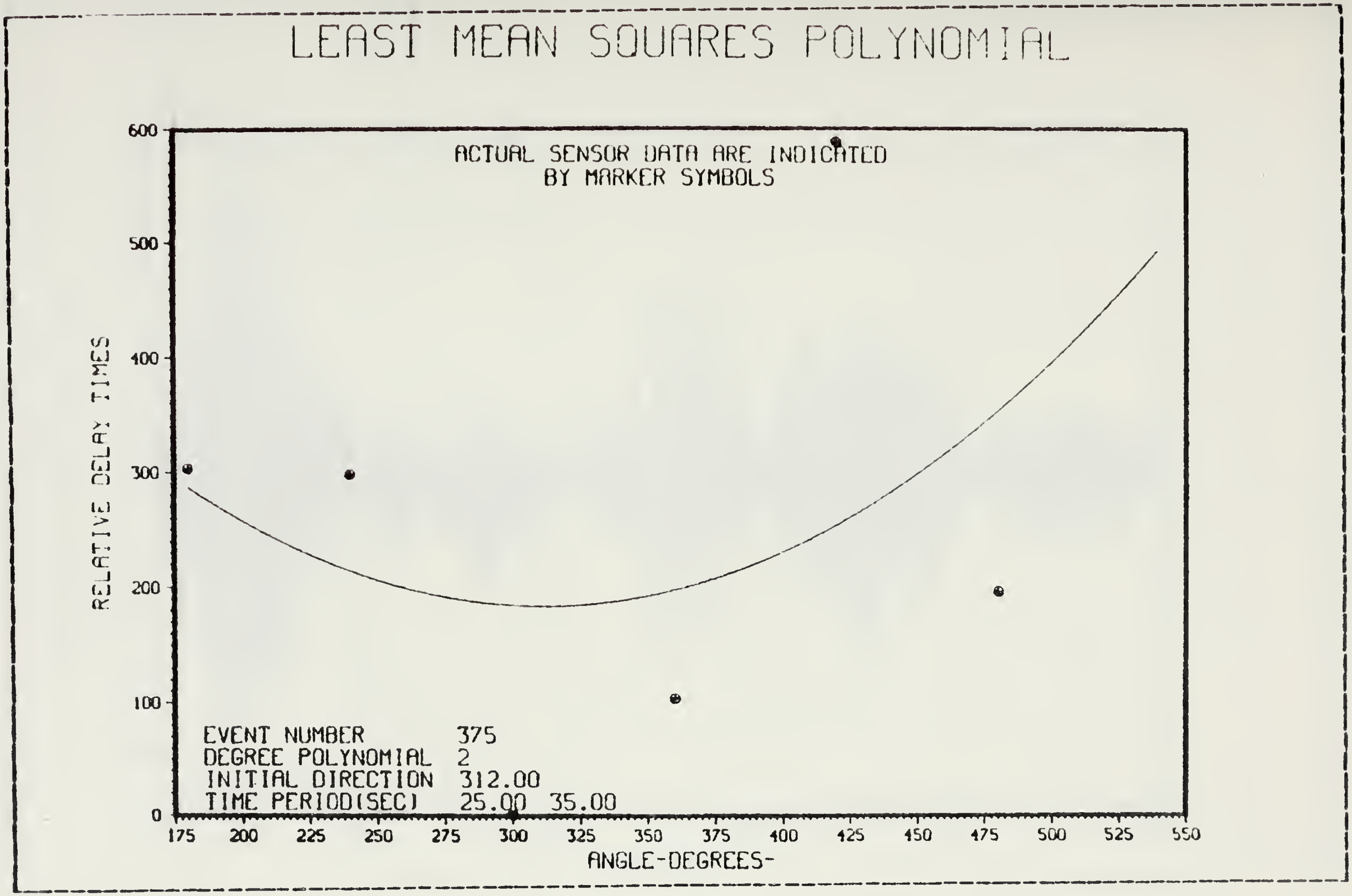

Figure 6.33 LMSP Initial Direction for Event 375

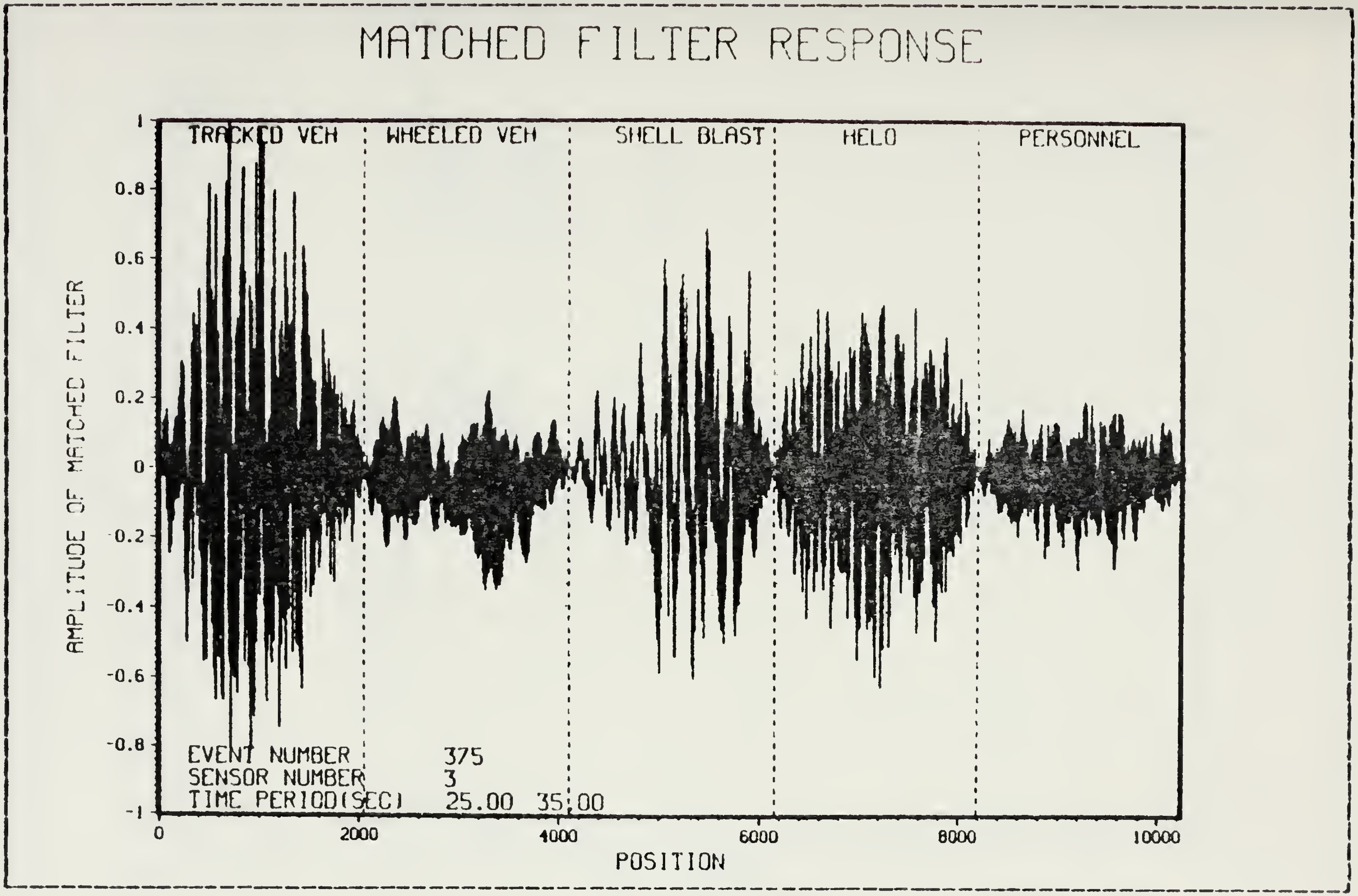

Figure 6.34 Matched Filter Response for Event 375

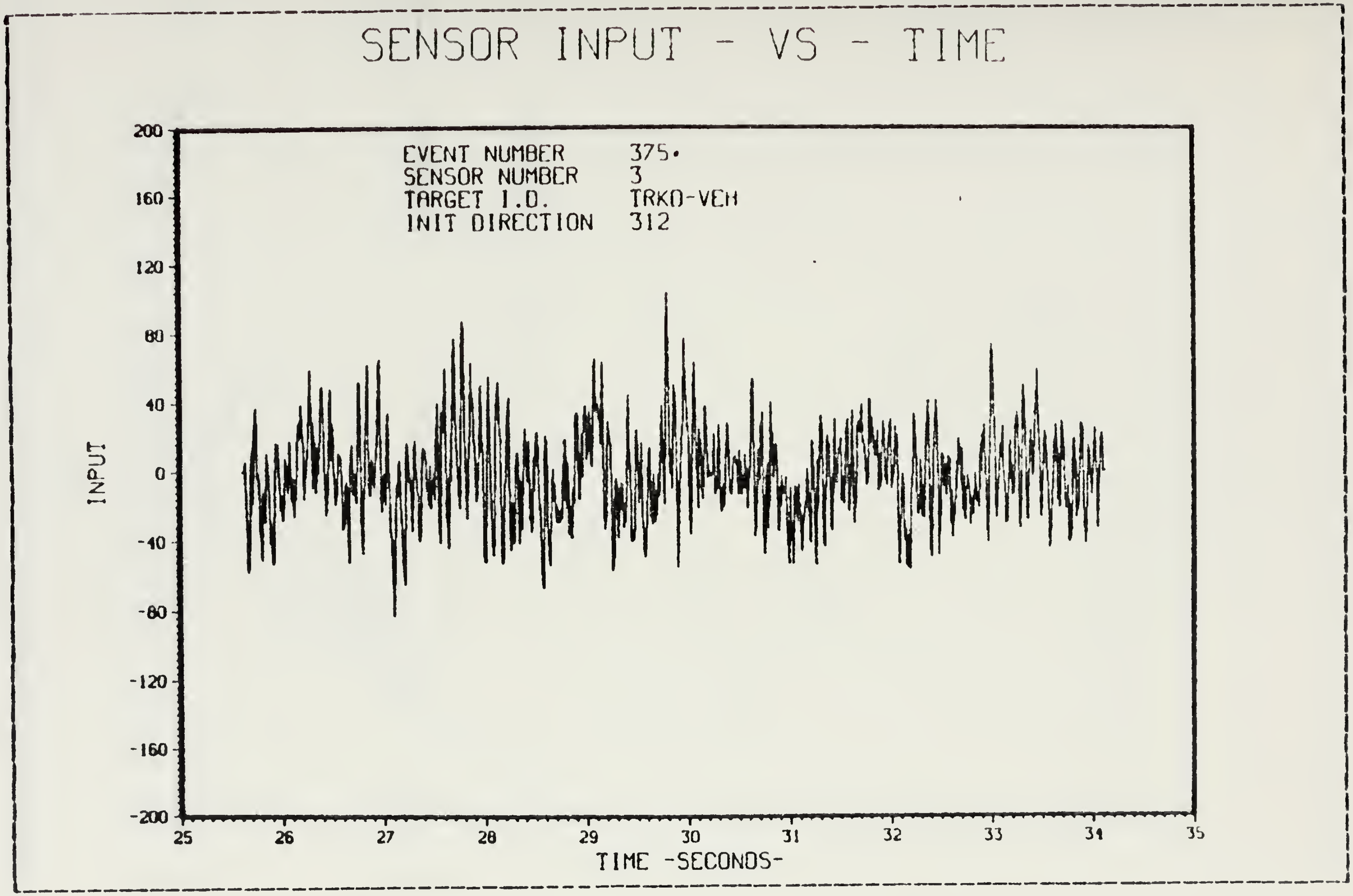

Figure 6.35 Amplitude Response for Event 375

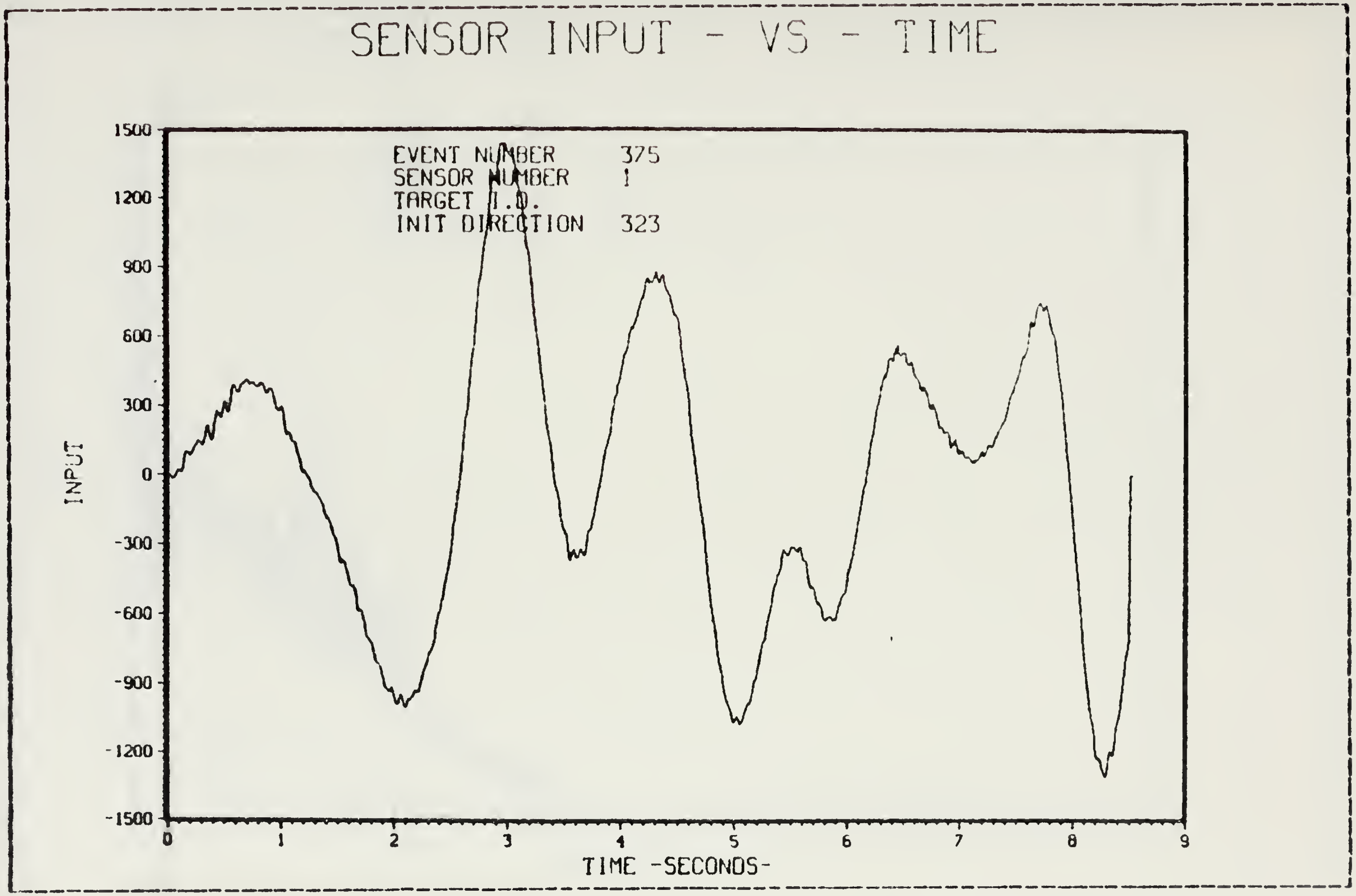

Figure 6.36 Amplitude Response of Malfunctioning Sensor for Event 375

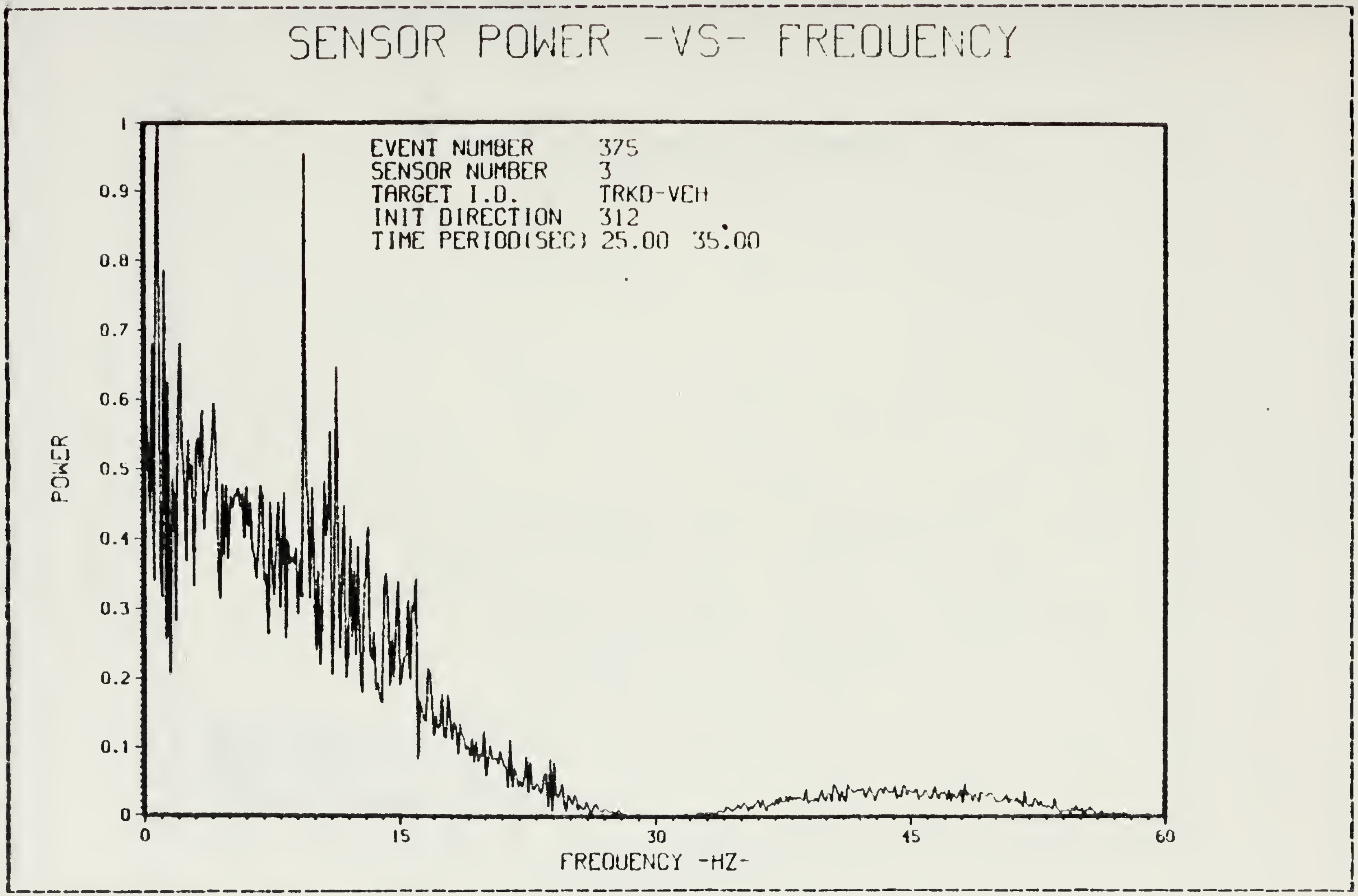

Figure 6.37 Frequency Response for Event 375

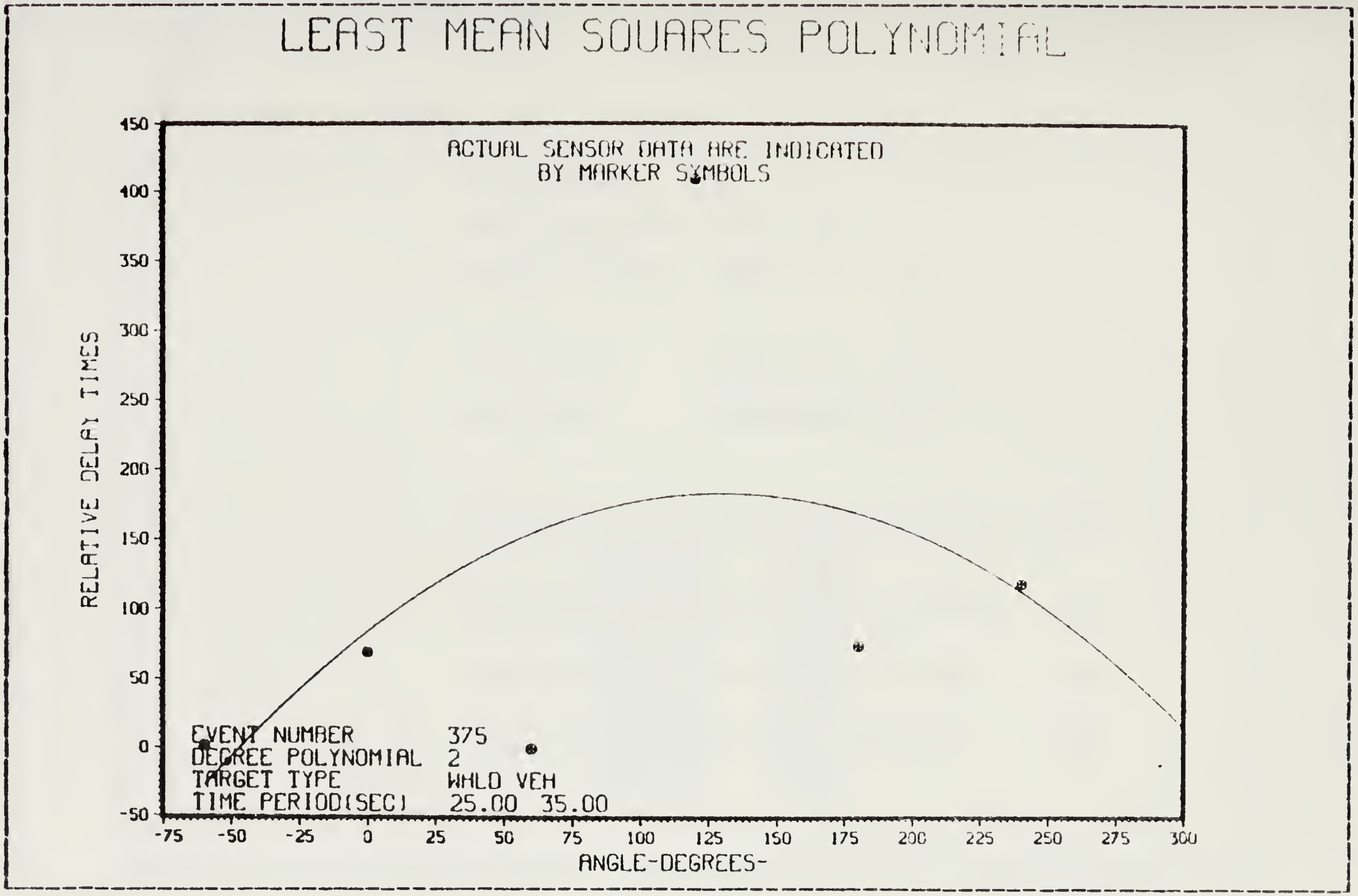

Figure 6.38 LMSP Matched Filter Direction for Event 375

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```
EVENT NUMBER     375

TIME PERIOD(SEC) 25.00  35.00

TRACKED VEHICLE   DIRECTION - 0.00


SHELL BLAST       DIRECTION - 312.00

HELICOPTER      . DIRECTION - 0.00



SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 6.39   LMSP Multiple Target Direction Summary for Event 375

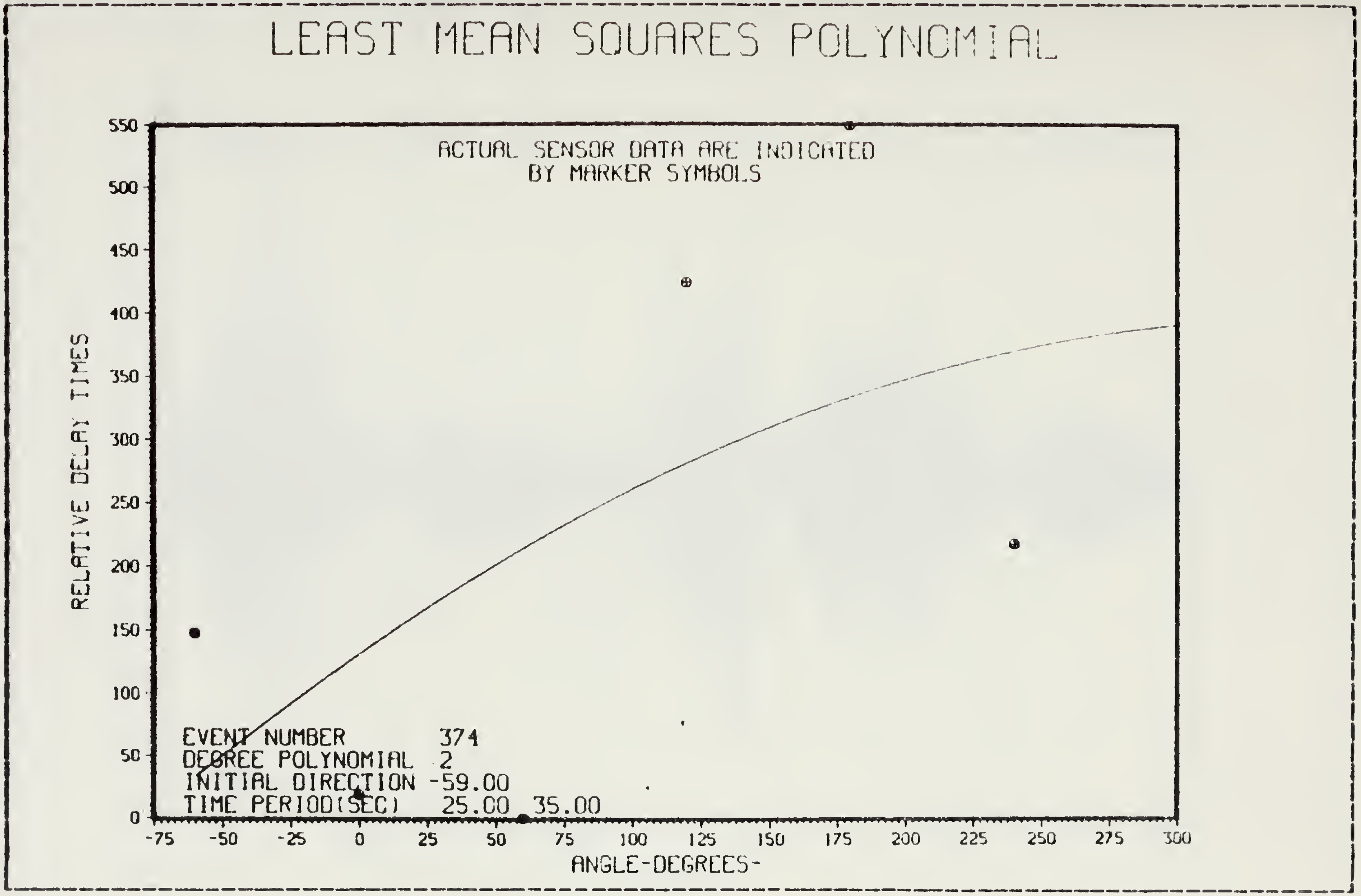

Figure 6.40 LMSP Initial Direction for Event 374

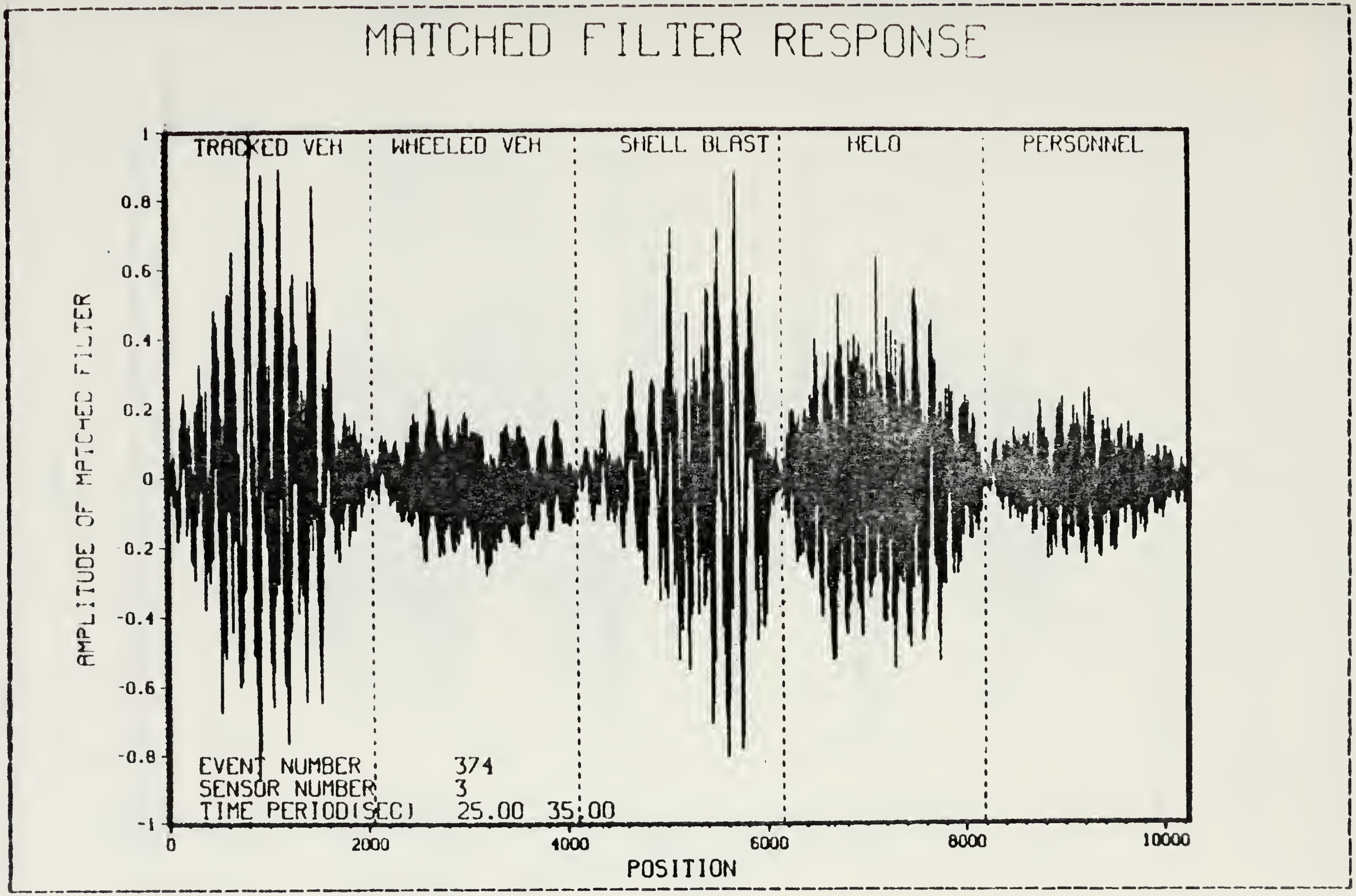

Figure 6.41 Matched Filter Response for Event 374

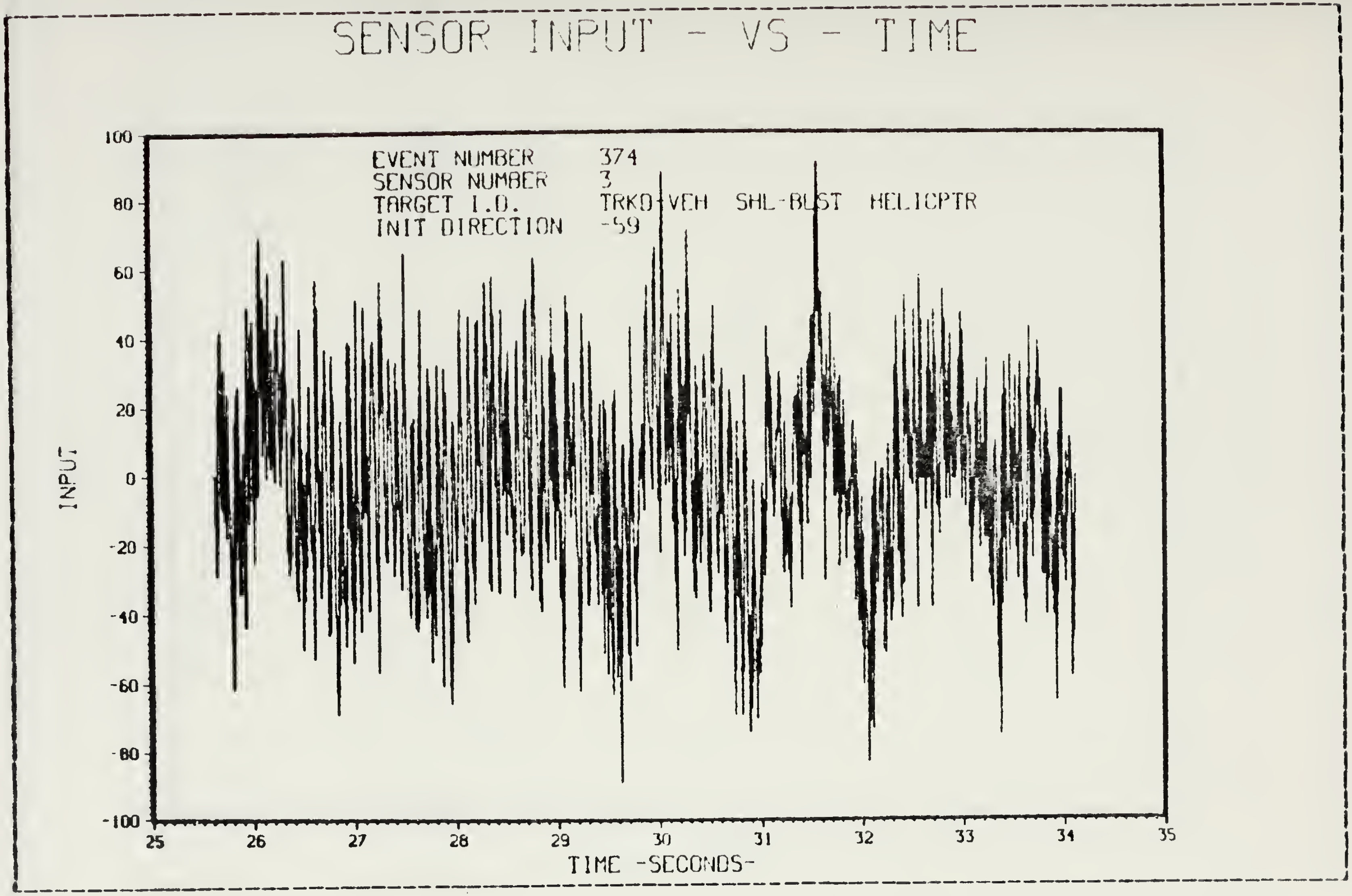

Figure 6.42 Amplitude Response for Event 374

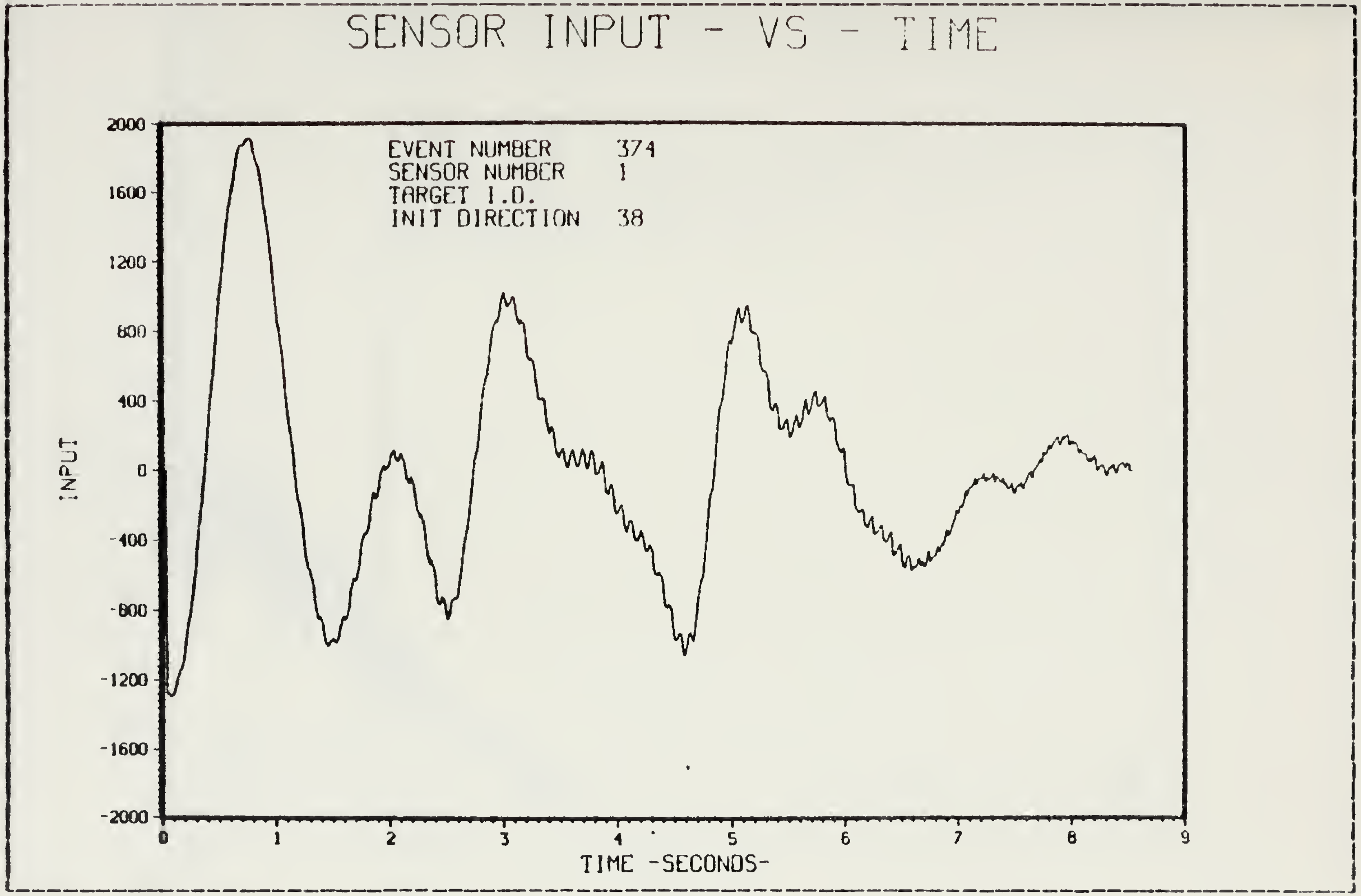

Figure 6.43 Amplitude Response of Malfunctioning Sensor for Event 374

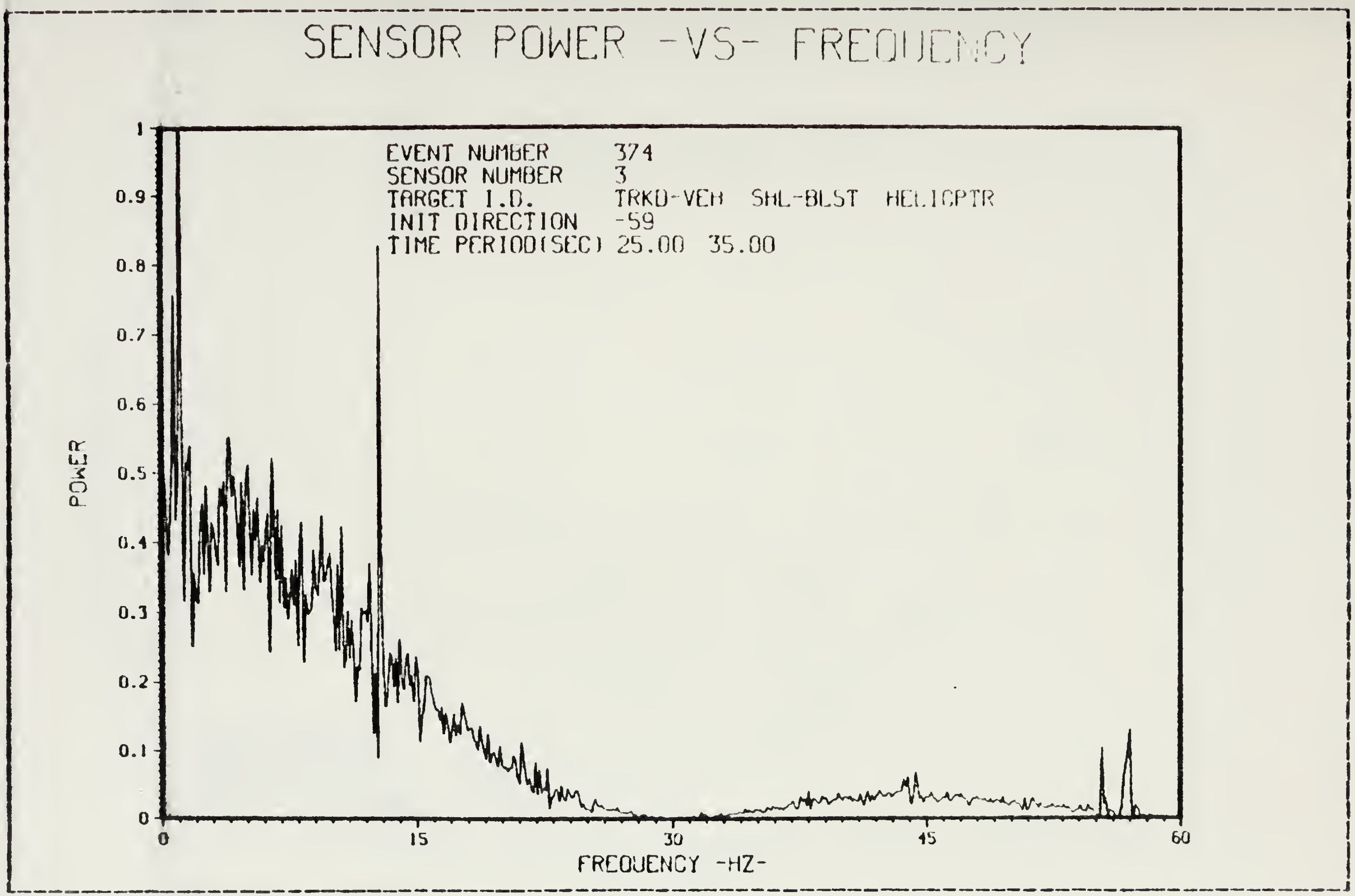

Figure 6.44 Frequency Response for Event 374

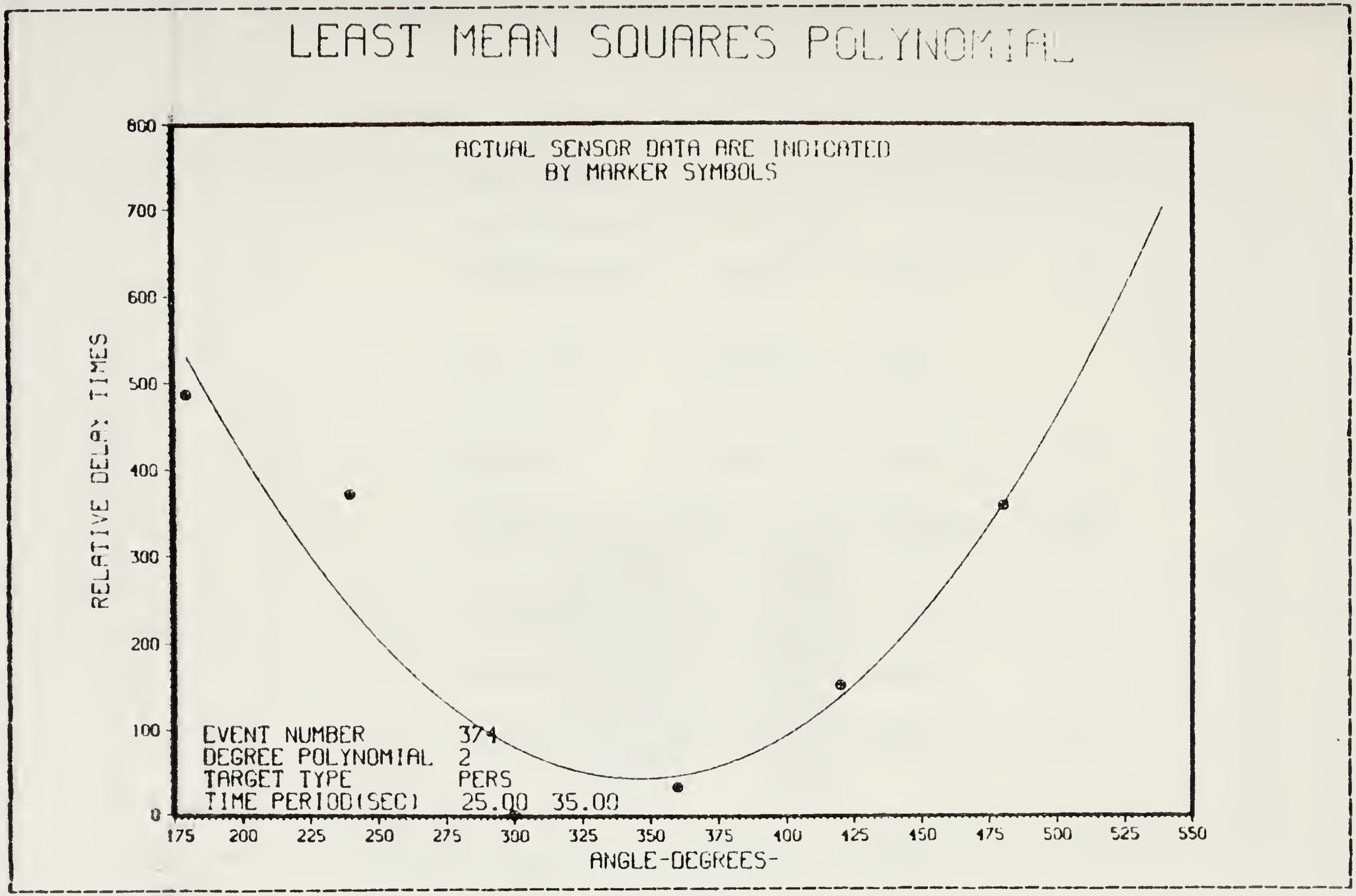

Figure 6.45 LMSP Matched Filter Direction for Event 374

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```
EVENT NUMBER     374

TIME PERIOD(SEC) 25.00  35.00

TRACKED VEHICLE    DIRECTION - -59.00


SHELL BLAST        DIRECTION - -59.00


PERSONNEL          DIRECTION - 347.00


SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
          AMPLITUDE     0.0000
          DIRECTION     0.0000
```

Figure 6.46 LMSP Multiple Target Direction Summary for Event 374

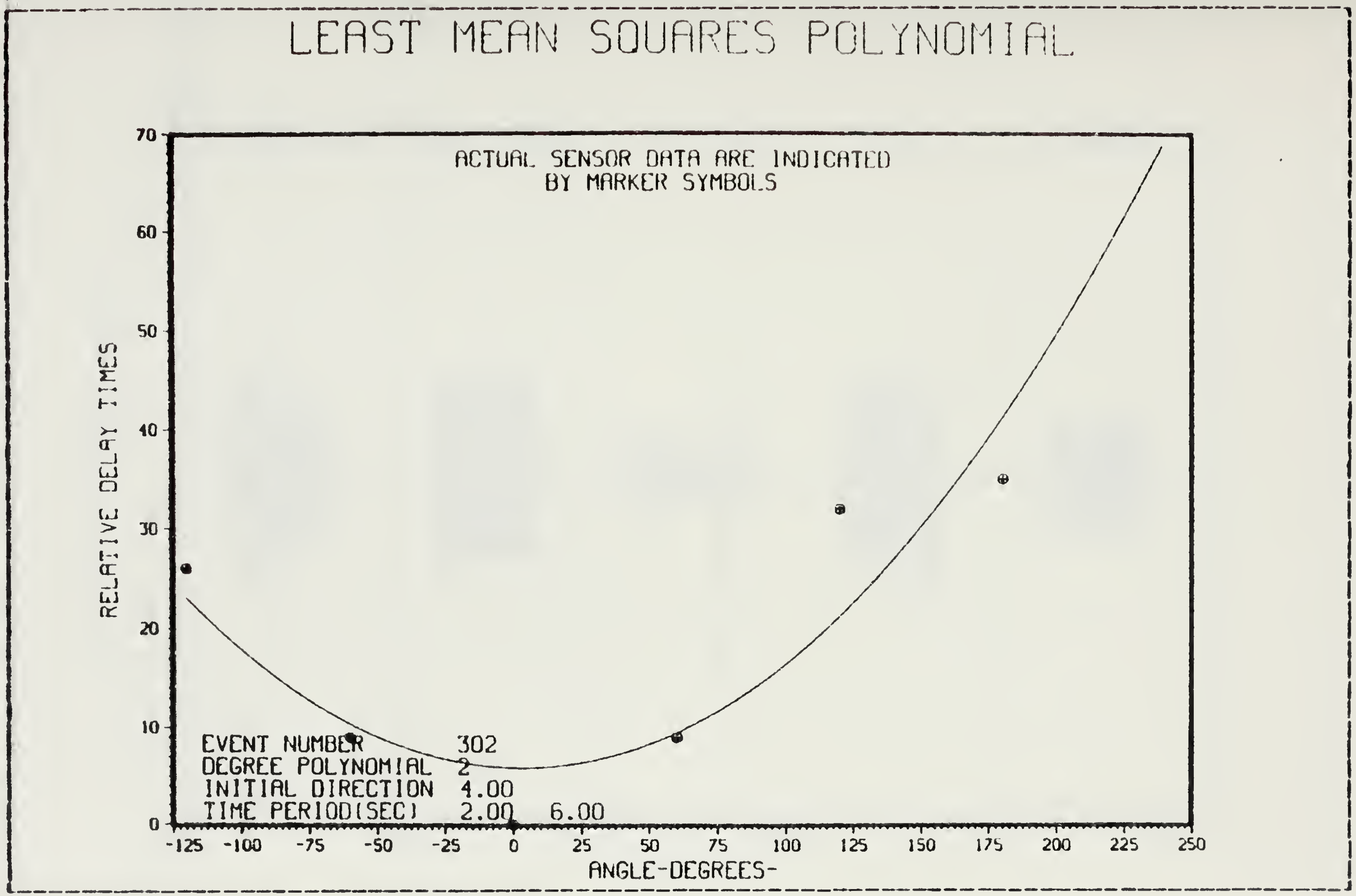

Figure 6.47 LMSP Initial Direction for Event 302

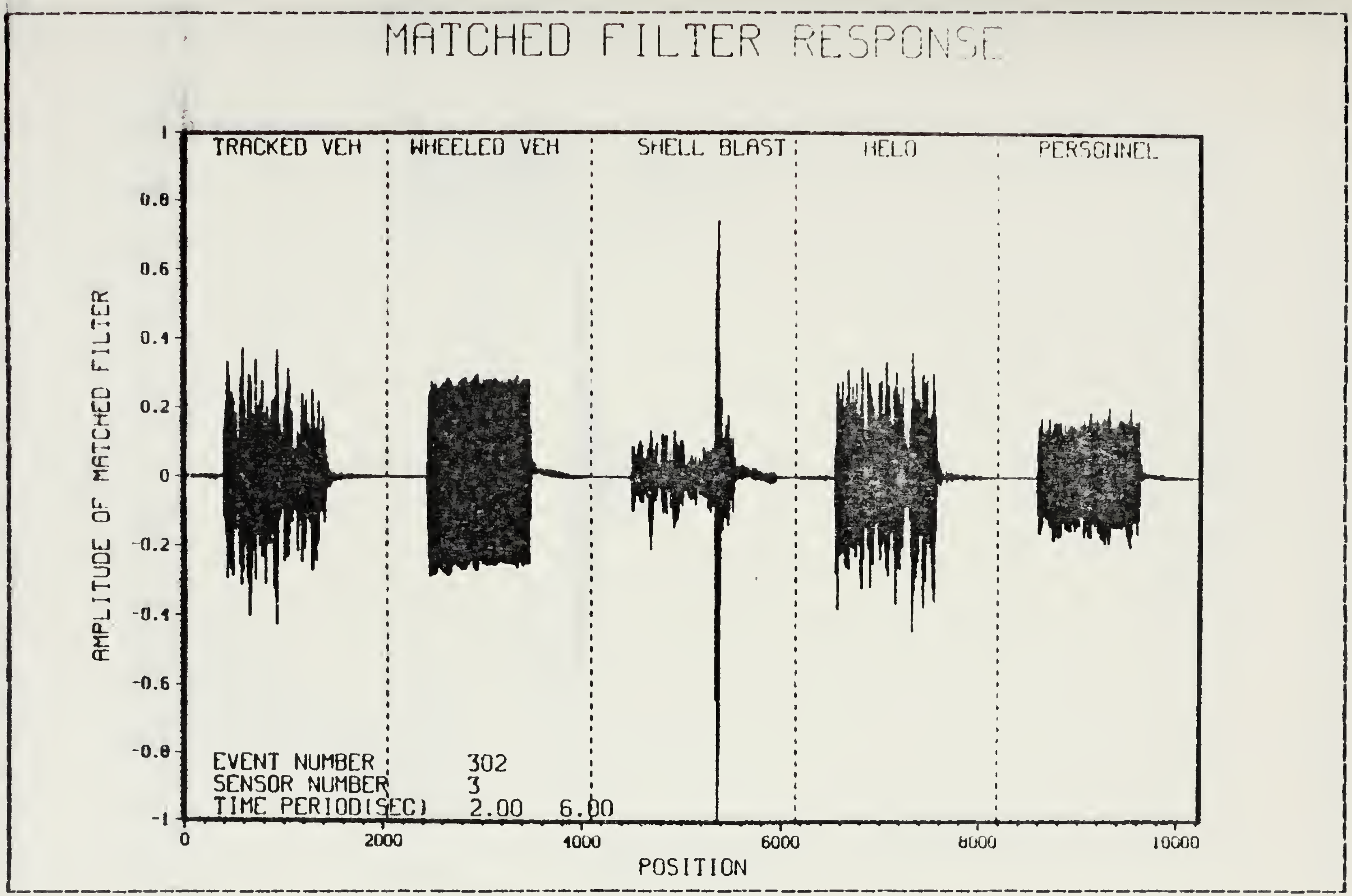

Figure 6.48 Matched Filter Response for Event 302

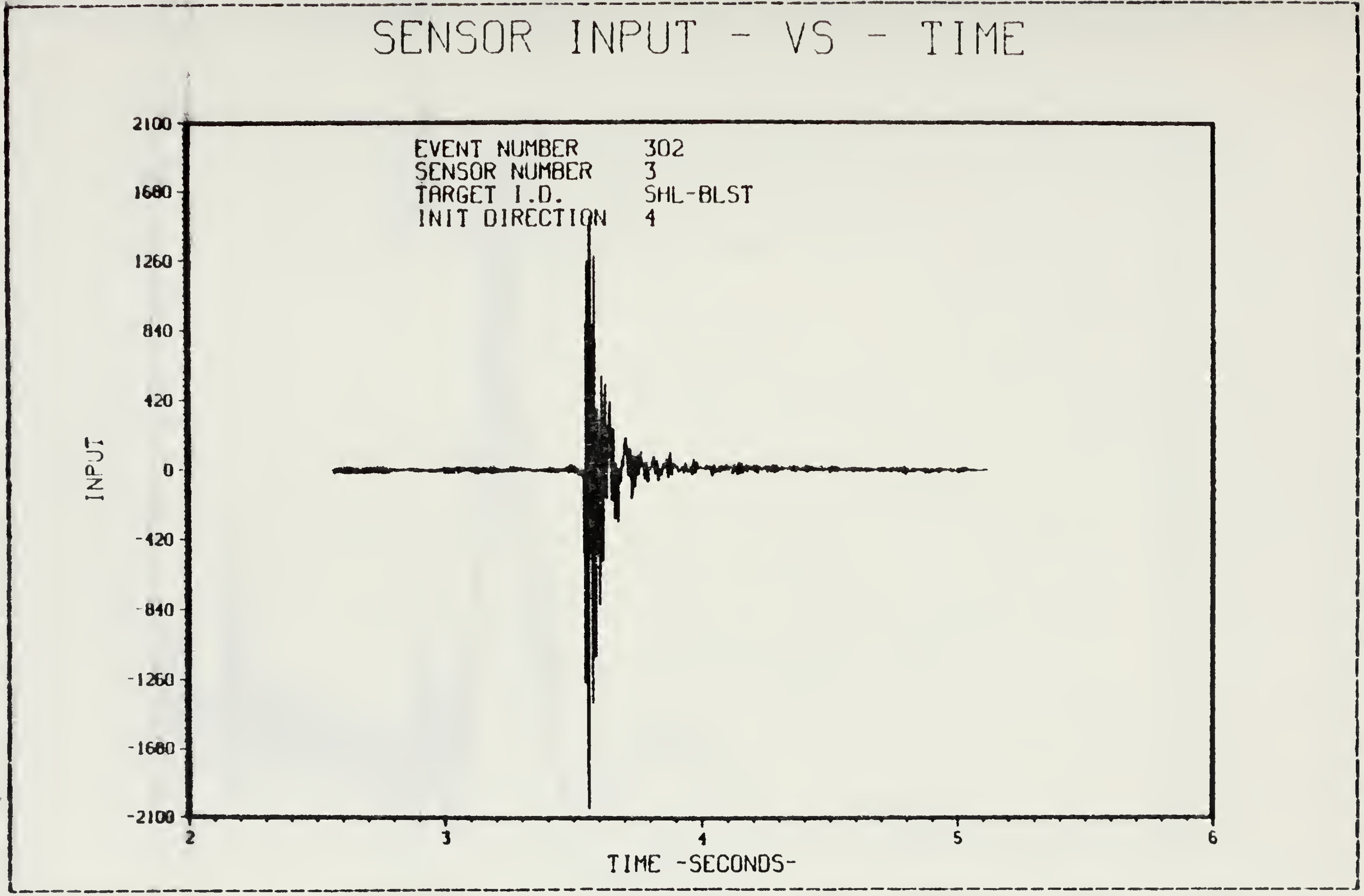

Figure 6.49 Amplitude Response for Event 302

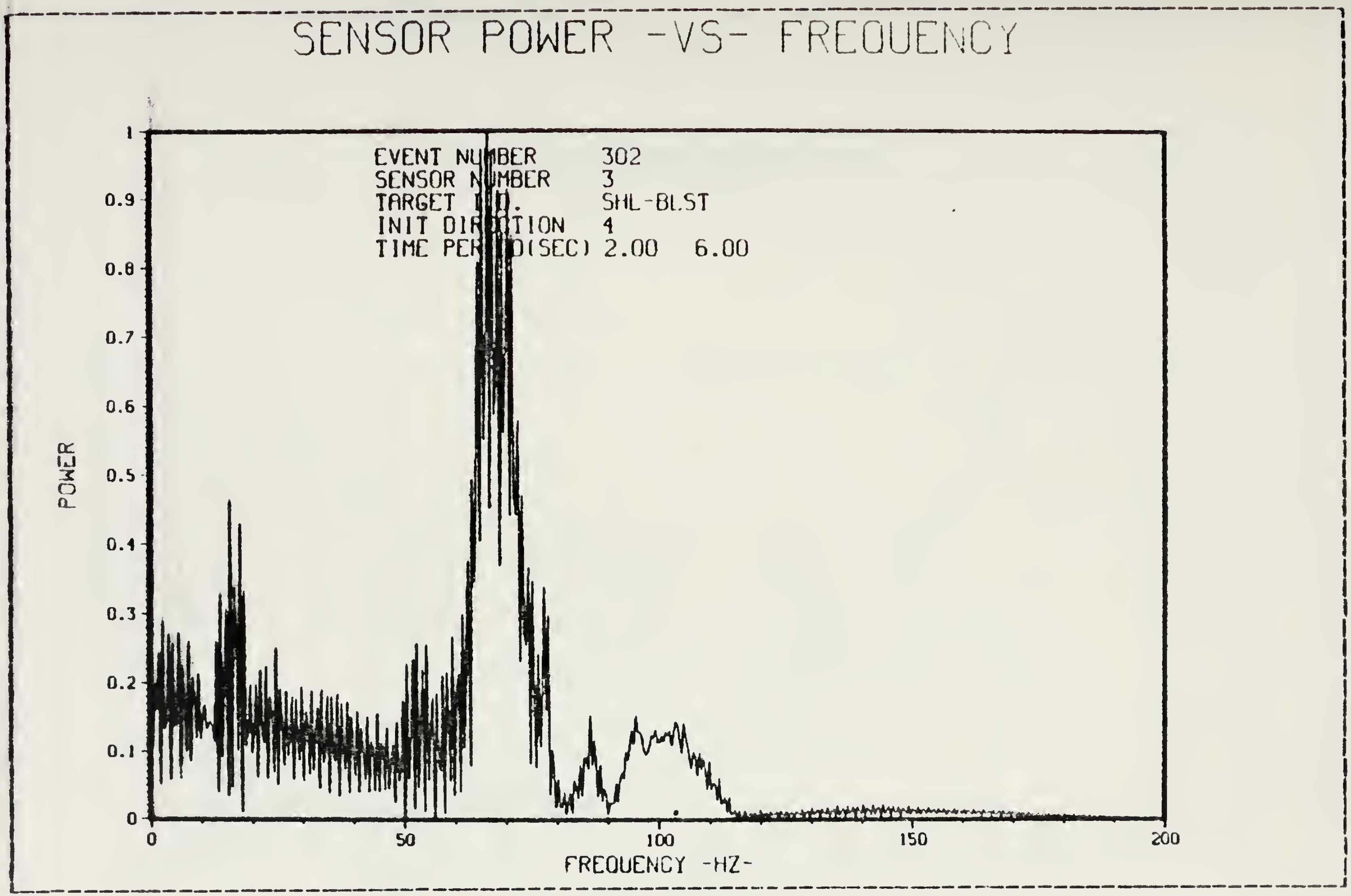

Figure 6.50 Frequency Response for Event 302

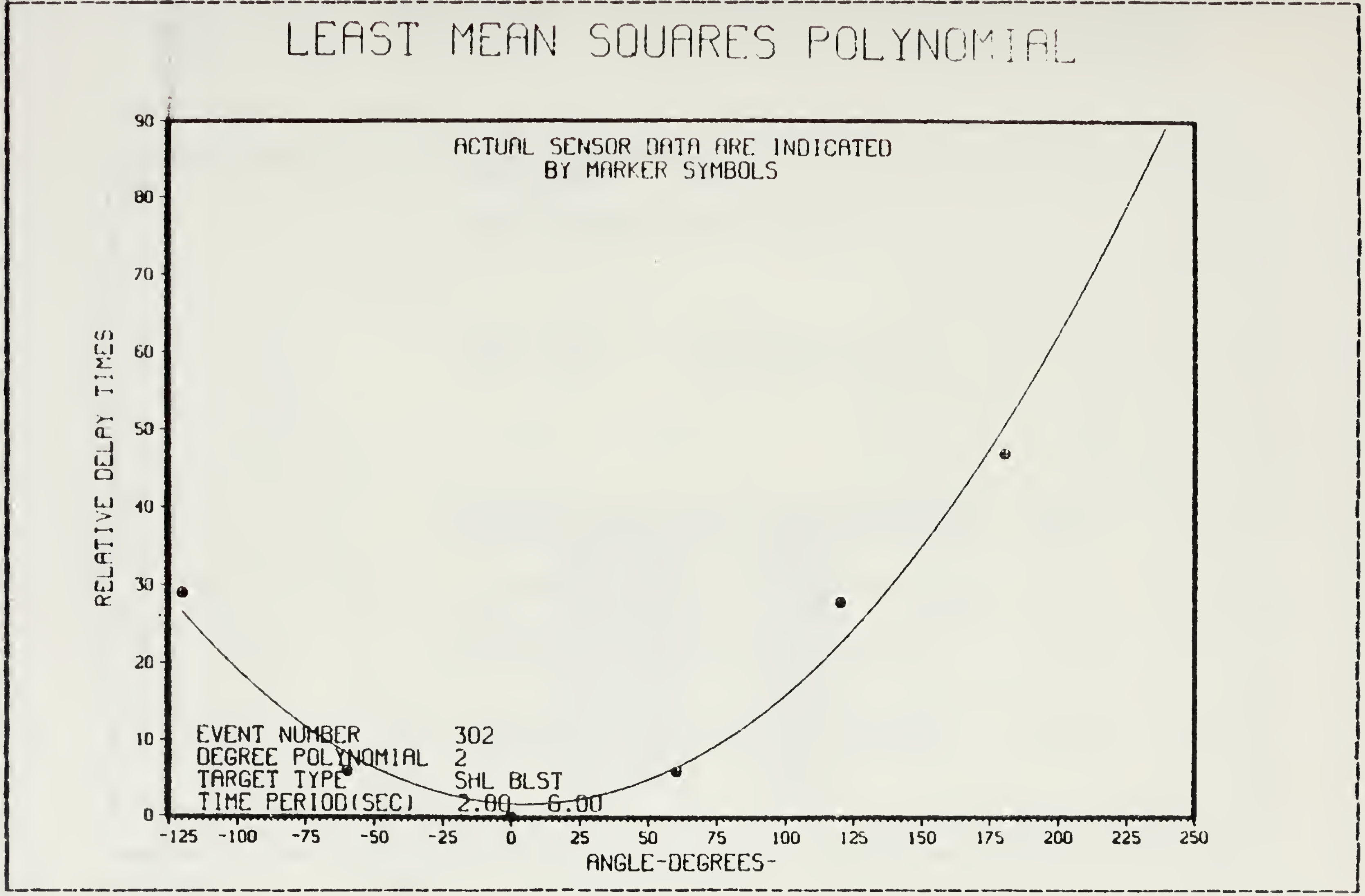

Figure 6.51 LMSP Matched Filter Direction for Event 302

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```
EVENT NUMBER      302

TIME PERIOD(SEC) 2.00   6.00


SHELL BLAST         DIRECTION - 6.00


SIMULATED TRKD VEHICLE TARGET FREQUENCY     0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY     0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY        0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 6.52 LMSP Multiple Target Direction Summary for Event 302

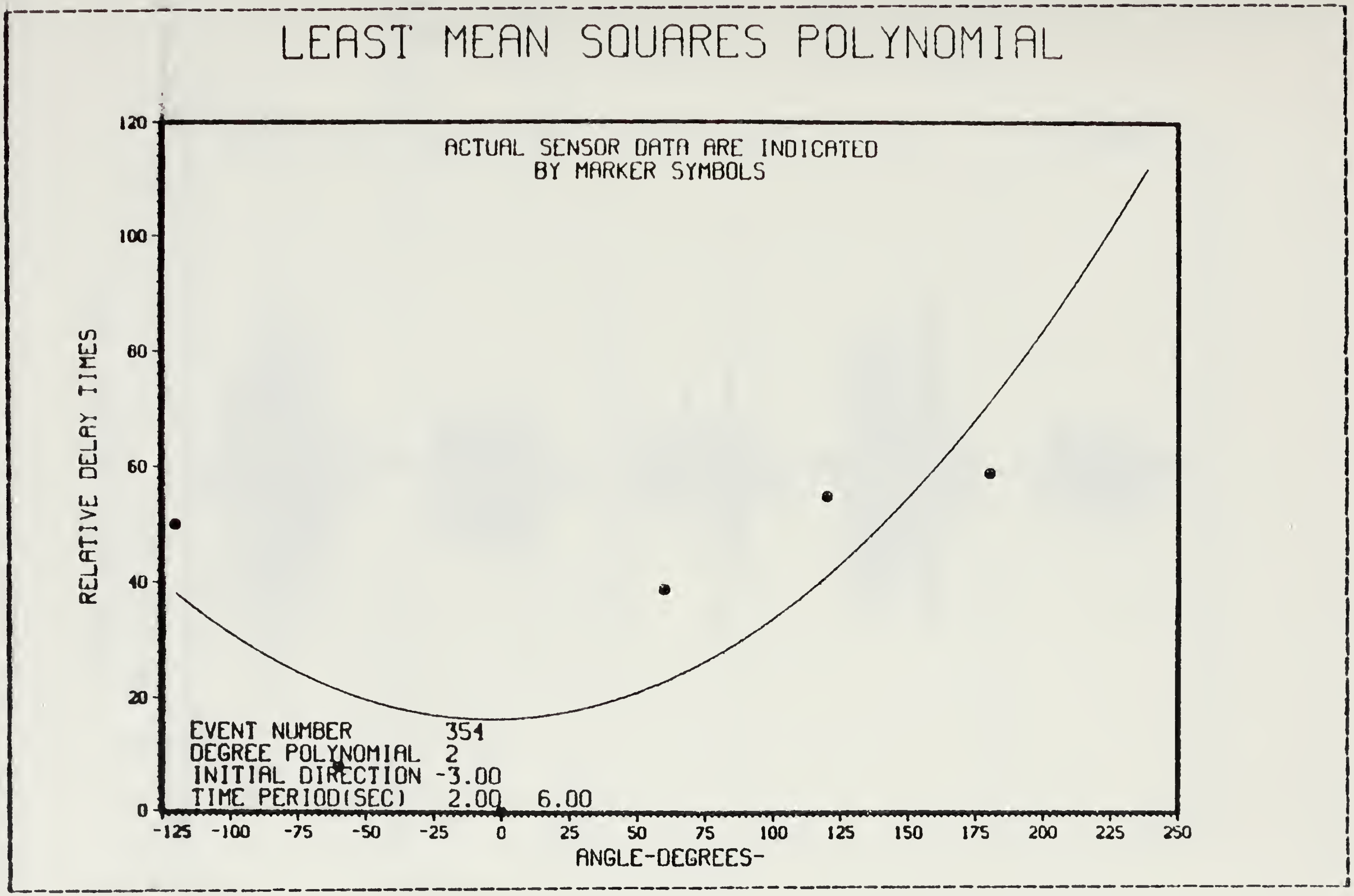

Figure 6.53 LMSP Initial Direction for Event 354 (2 - 6sec)

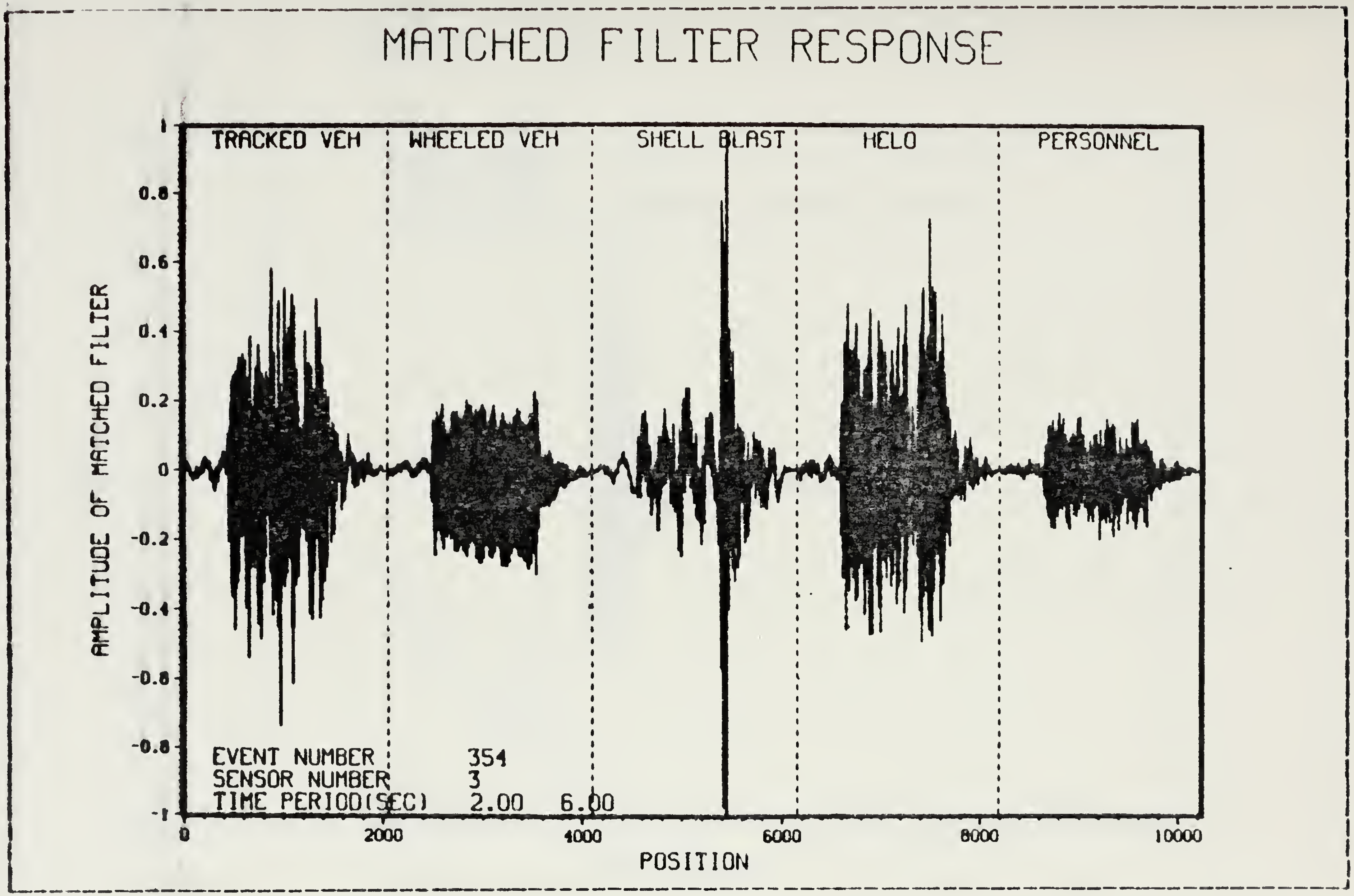

Figure 6.54 Matched Filter Response for Event 354 (2 - 6sec)

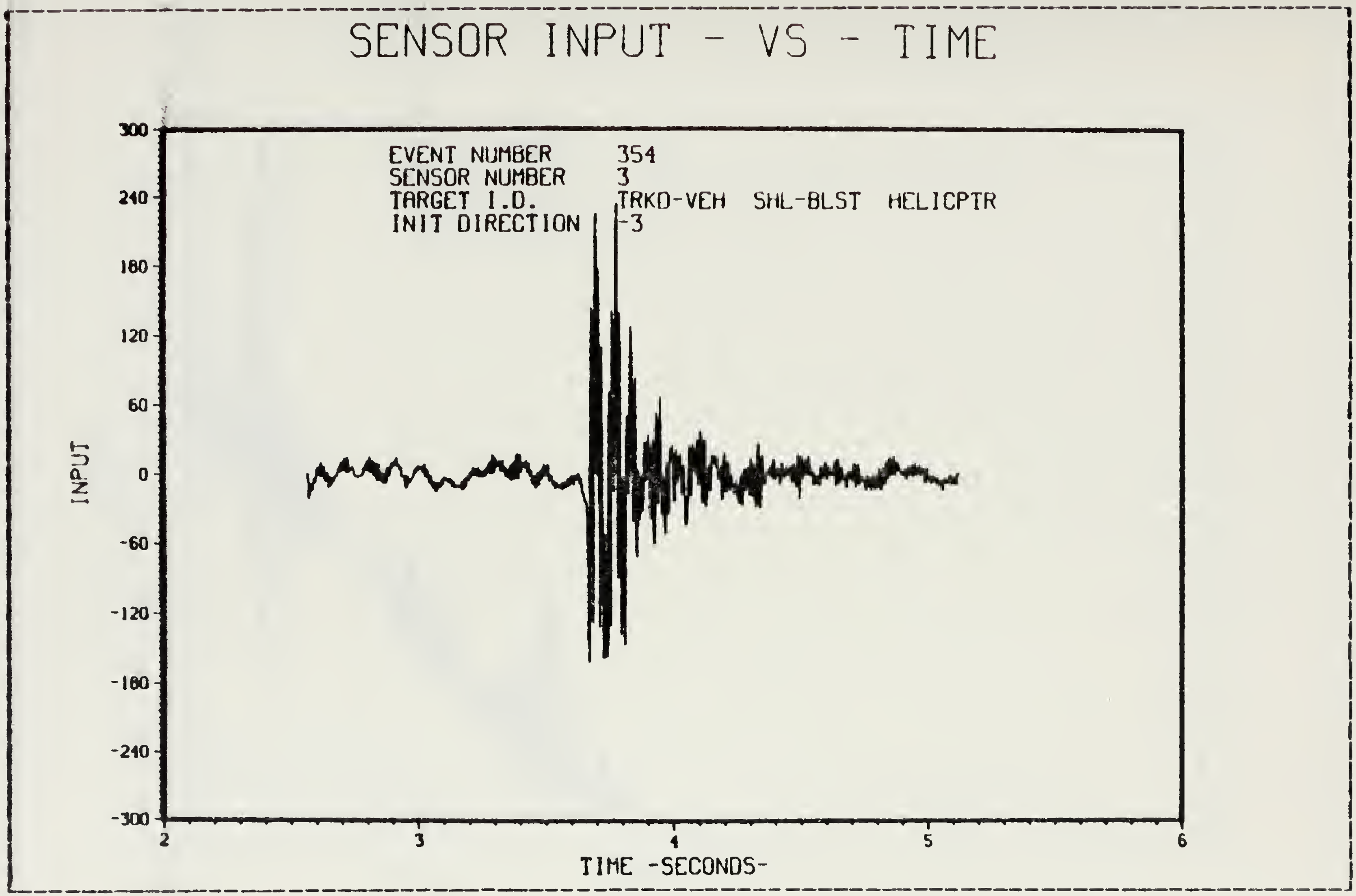

Figure 6.55 Amplitude Response for Event 354 (2 - 6sec)

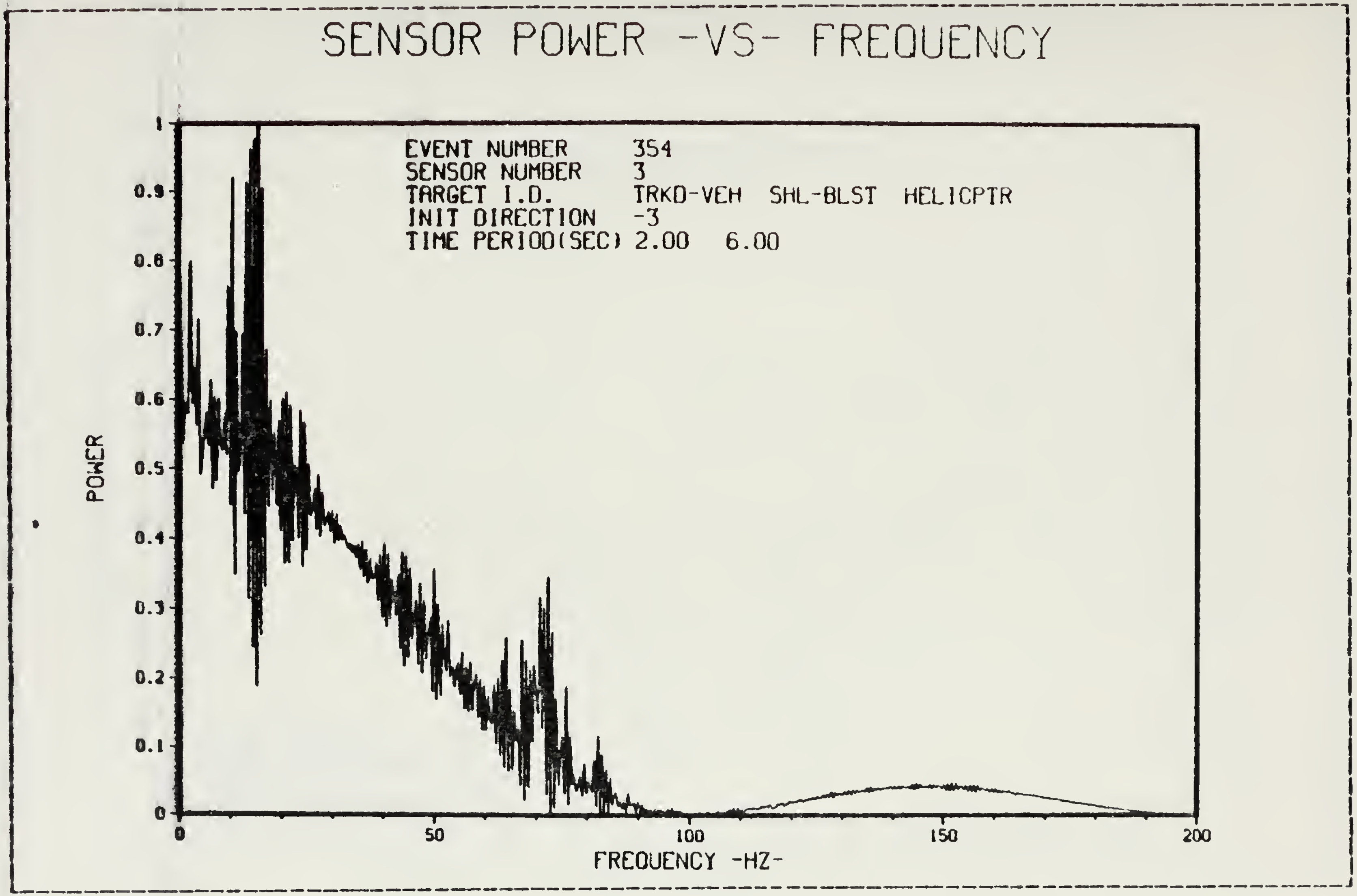

Figure 6.56 Frequency Response for Event 354 (2 - 6sec)

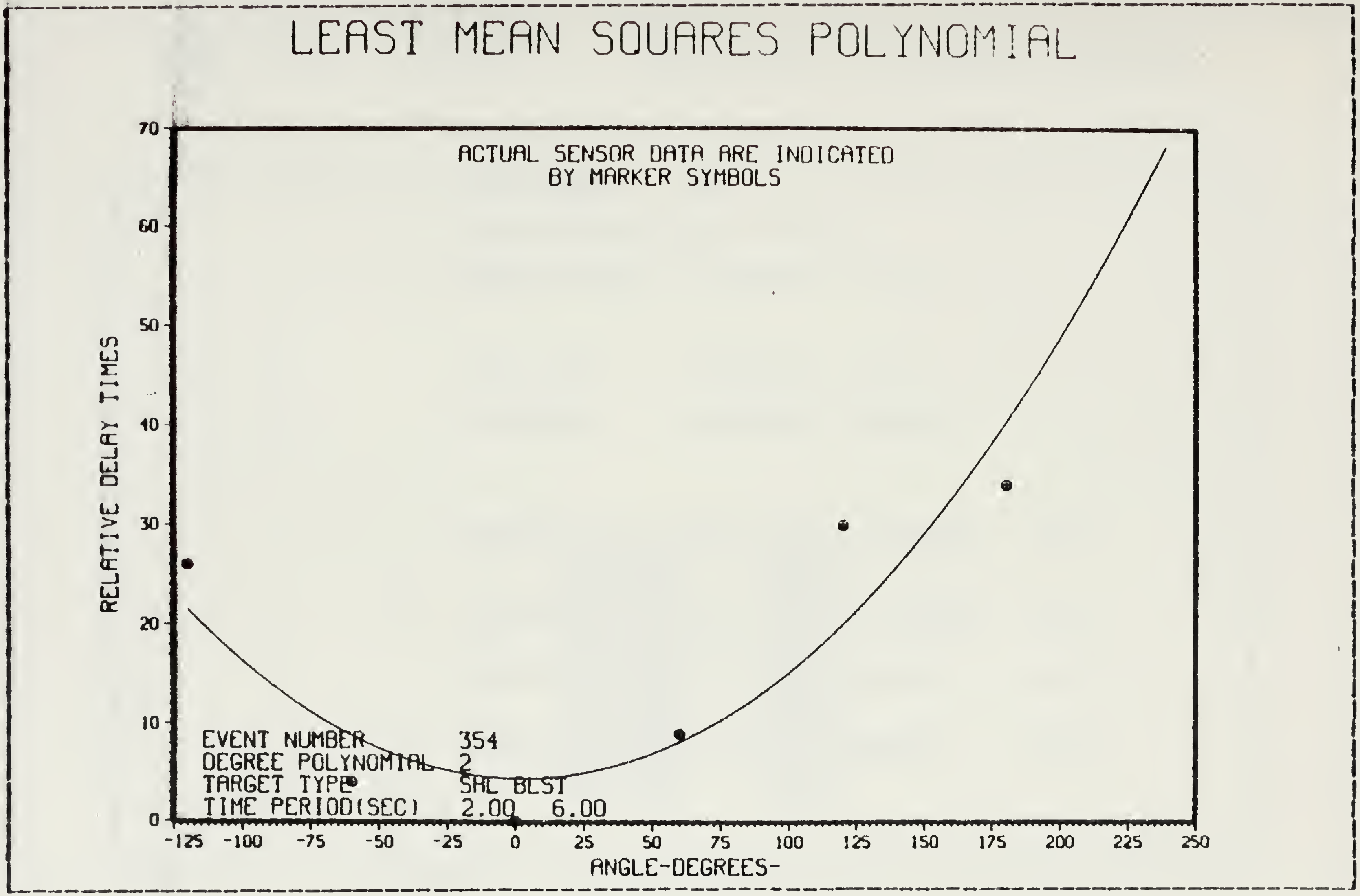

Figure 6.57 LMSP Matched Filter Direction for Event 354 (2 - 6sec)

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```
EVENT NUMBER     354

TIME PERIOD(SEC) 2.00   6.00

TRACKED VEHICLE    DIRECTION - 332.00


SHELL BLAST        DIRECTION - 3.00

HELICOPTER         DIRECTION - 355.00



SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 6.58   LMSP Multiple Target Direction Summary Event 354 (2 - 6sec)

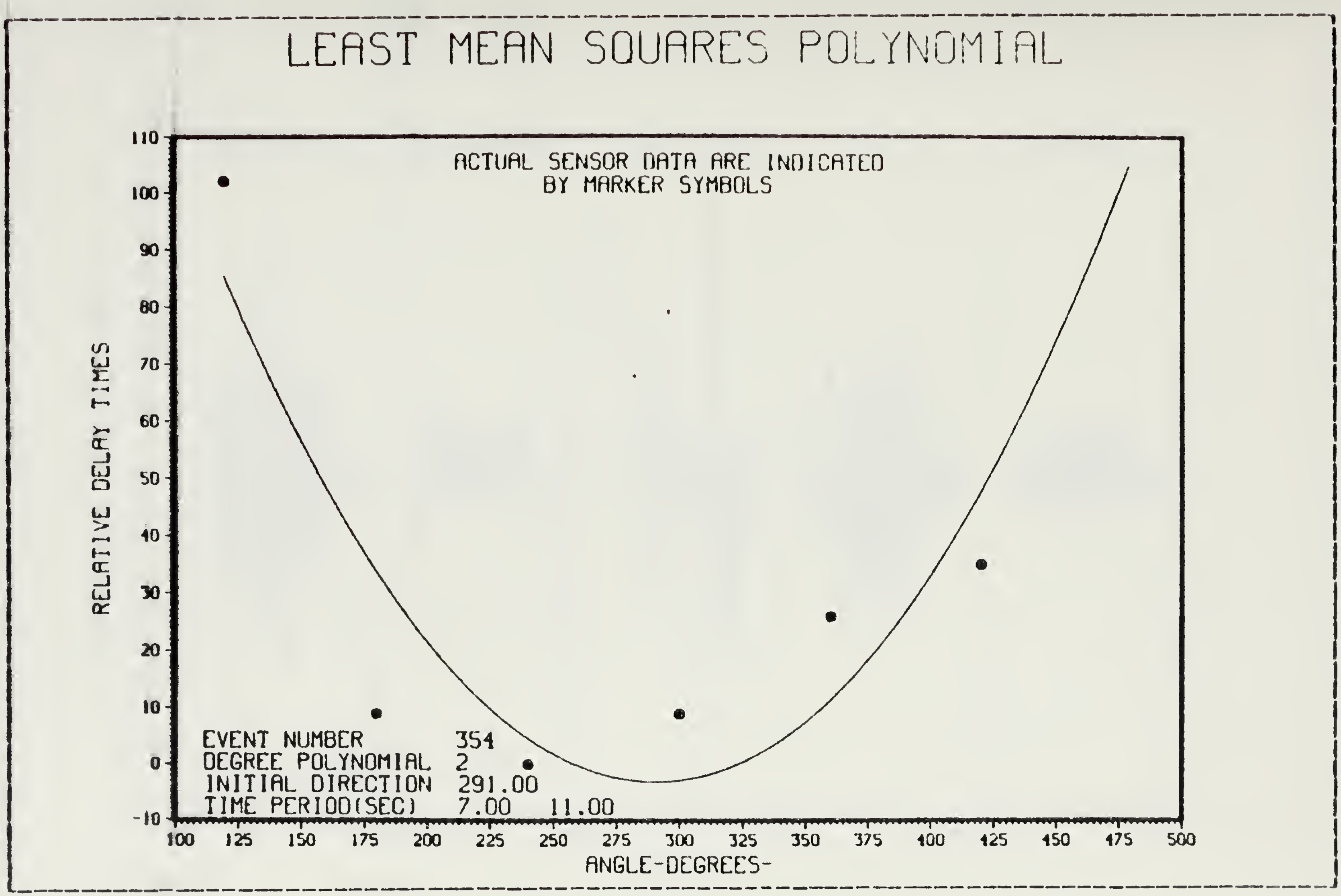

Figure 6.59 LMSP Initial Direction for Event 354 (7 - 11sec)

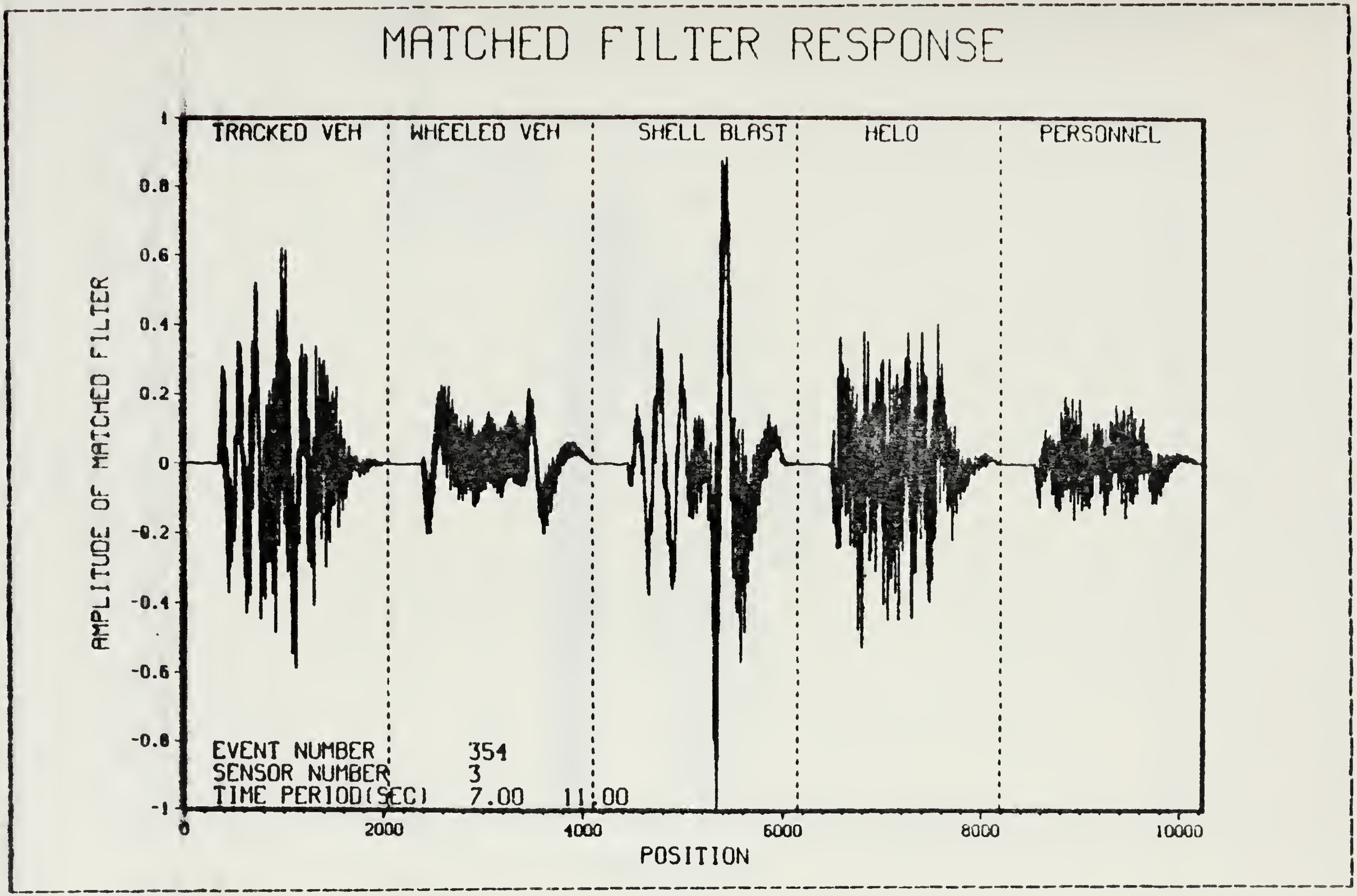

Figure 6.60 Matched Filter Response for Event 354 (7 - 11sec)

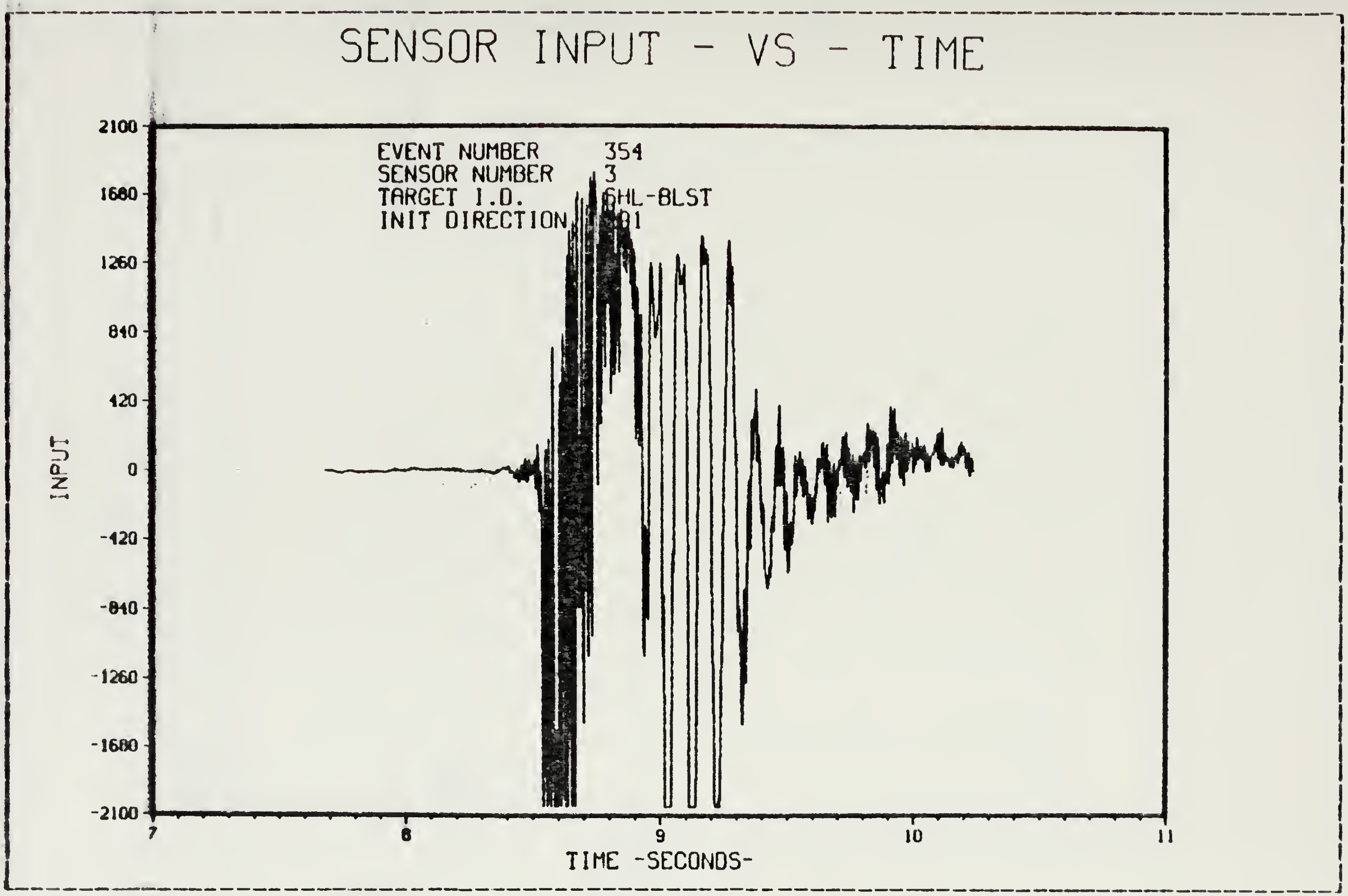

Figure 6.61 Amplitude Response for Event 354 (7 - 11sec)

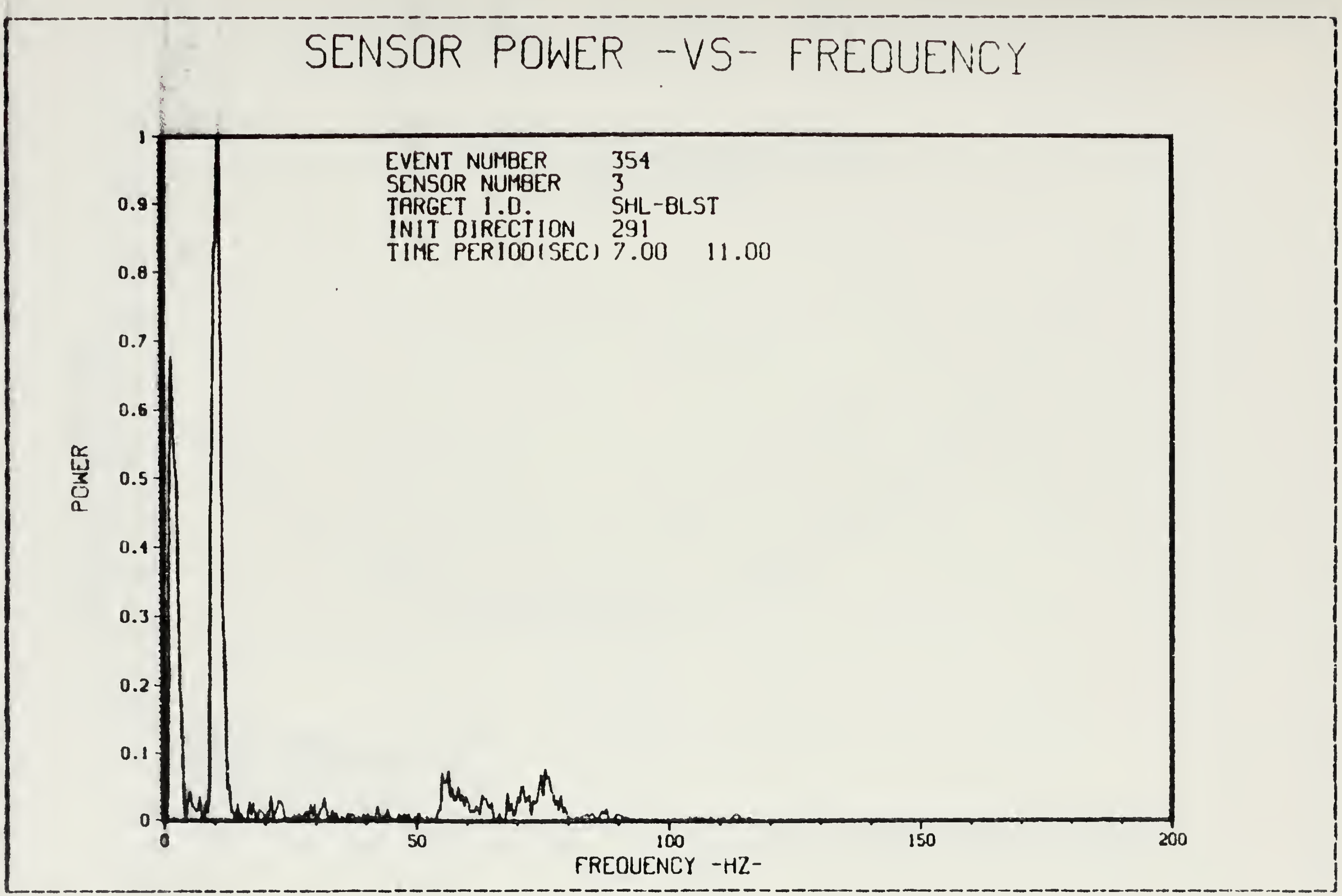

Figure 6.62 Frequency Response for Event 354 (7 - 11sec)

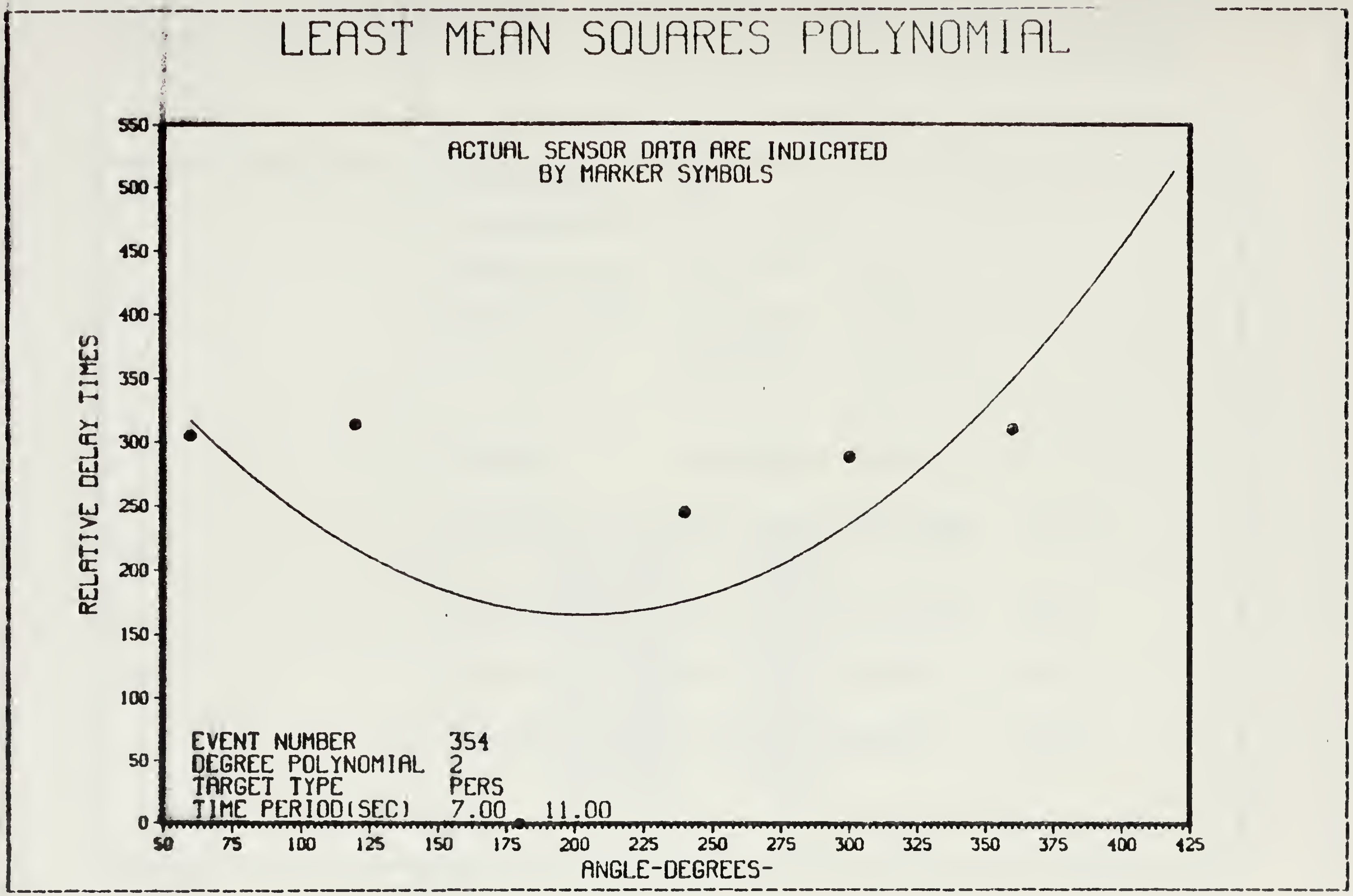

Figure 6.63 LMSP Matched Filter Direction for Event 354 (7 - 11sec)

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```
EVENT NUMBER     354

TIME PERIOD(SEC) 7.00   11.00

TRACKED VEHICLE   DIRECTION - 291.00

WHEELED VEHICLE   DIRECTION - 291.00

SHELL BLAST       DIRECTION - 291.00


PERSONNEL         DIRECTION - 204.00

SIMULATED TRKD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED WHLD VEHICLE TARGET FREQUENCY    0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
SIMULATED PERSONNEL TARGET FREQUENCY       0.00
        AMPLITUDE     0.0000
        DIRECTION     0.0000
```

Figure 6.64   LMSP Multiple Target Summary Event 354 (7 - 11sec)

VII. CONCLUSIONS AND RECOMMENDATIONS

The ability of the digital matched filter to detect and correctly identify actual discrete single target types was excellent. The adaptive enhancement of the matched filter scheme was found to sharpen the matched filter responses. The complications of multiple targets and continuous targets proved to be less successful. Lack of high signal to noise ratio sample signals for use as filters reduced the abiltiy of the filters to match the signals. As would be expected, the simulated target identification and direction finding operations met with success for both single and multiple targets.

The simulated data validated the usefulness of the least mean squares curve fitting method for target direction finding. This algorithm was noted to be useful in both the relative peak amplitude response method for recoil/blast targets and the matched filter peak position method for all targets. The highest accurracies were found using second degree polynomials. This was due to the reduced noise sensitivity of lower degree polynomials. Conversly, significant errors were found in the directions determined by the phase difference algorithm. These errors were possibly due to round-off error sensitivity in the software/hardware implementation. Experimental data could not be used to effectively crosscheck this finding since most of the experimental data targets were at zero degrees and the phase difference routine seemed to seek zero degrees.

A significant result was the accuracy of the blast/recoil target direction found using only the peak amplitude responses. Directions could be found using only the relative amplitude peak positions for the array sensors and the

least mean squares curve fitting routine. This result indicated a possible counter-fire application using a greatly simplified system. This finding is felt to be significant since artillery type targets are the highest priority target type. The discrete nature of the blast/recoil seismic signals would also allow for ready separation of even a large number of combined hostile and friendly signal sources. This would allow for observerless adjustment of fire onto hostile targets. Artillery and mortar targets may, infact, be the only target types detectable at the ranges specified for a long range seismic system.

It is recommeded that further study be made in to the possible implementation of the least mean square curve fitting of the peak sensor amplitudes responses in a counter-fire system. Digital matched filters for seismic target identification and direction finding may also prove effective after further experimentation with optimum sample signal filters for the various target types. Additionally, the matched filter response may possibly be enhanced by preprocessing the seismic signals through adaptive noise cancellors.

Acoustic vice seismic matched filtering with directional microphones may be useful in target identification. Once the target has been identified, a matched filter/least mean squares based direction finding scheme could then be attempted.

APPENDIX A

USERS MANUAL

The software developed provided for interactive program operation. However, further information must be provided for an initial system setup and correct program operation. To begin with, the seismic data must be transferred from magnetic tape to the IBM 3033's Mass Storage System (MSS). The data must then be transfered to the DISSPLA user's disk for analysis of the program. These data transfers may be accomplished using Jcb Control Language (JCL) procedures.

The magnetic tape volume must first be scanned to determine the storage format of its files. The JCL procedure TSCAN provides this information. A sample TSCAN job follows:

```
//JLJV1677 JOB (3026,0304),'SMC-1677 JOHNSTON',CLASS=F
//  EXEC TSCAN,VOLIN=PARK1,DCBIN='DEN=2',UNITIN='3400-4'
//  EXEC TSCAN,VOLIN=PARK2,DCBIN='DEN=2',UNITIN='3400-4'
//  EXEC TSCAN,VOLIN=PARK3,DCBIN='DEN=2',UNITIN='3400-4'
//  EXEC TSCAN,VOLIN=PARK4,DCBIN='DEN=2',UNITIN='3400-4'
//
```

Once the tape scan is completed, the tape files and comments can be transferred to the MSS. Prior to this transfer however, space in the MSS must be made for these files. The procedure IEFBR14 is used for this purpose. A sample job follows:

```
//JLJTM74A JOB (3026,0304),'SMC1677 JOHNSTON',CLASS=A
//*MAIN ORG=NPGVM1.0131P
//  EXEC PGM=IEFBR14
//DD1 DD UNIT=3330V,MSVGP=PUB4B,DISP=(NEW,CATLG),
//         SPACE=(CYL,(4,4,3)),DSN=MSS.S3026.P302
```

```
/*
//  EXEC PGM=IEFBR14
//DD1 DD UNIT=3330V,MSVGP=PUB4B,DISP=(NEW,CATLG),
//        SPACE=(CYL,(4,4,3)),DSN=MSS.S3026.P314
/*
//  EXEC PGM=IEFBR14
//DD1 DD UNIT=3330V,MSVGP=PUB4B,DISP=(NEW,CATLG),
//        SPACE=(CYL,(4,4,3)),DSN=MSS.S3026.P319
/*
//
```

Each event is proceeded by a comment file for that event. These comment files can be identified from the TSCAN output as a file containing only one record. The files containing nine or eighteen records are the sensor data files for the events. Files of nine records in length are events using a circular array of nine sensors designated as a type A33 array. These sensors are all vertical motion-sensing geophones. The eighteen record files contain six sensor groups of three geophones. A circular array is designated type A31 and a linear array is a type A32 array. The three geophones for each group sense either radial, transverse or vertical motion. A JCL routine, using the procedure IEBGENER, transfers the event comment and sensor data. A sample IEBGENNER job follows:

```
//JLJ11677 JOB (3026,0304),'JOHNSTON SMC1677',CLASS=F
//*MAIN ORG=NPGVM1.0131P
//*
//* CP/CMS SUBMIT     IEBGENER JCL
//*
//*       COPY TAPE FILES TO MSS.S3026.P319
//*
//  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=3400-4,VOL=SER=PARK2,DISP=(,PASS),
```

```
//            LABEL=(1,NL,,IN),
//     DCB=(RECFM=FB,LRECL=64,BLKSIZE=2048,DEN=2,OPTCD=Q)
//SYSUT2 DD  DISP=SHR,DSN=MSS.S3026.COMMENTS(COM319)
//SYSIN DD DUMMY
//* * * * * * * * * * * * * * * * * * * * *
//COPY  PROC FILE=,MEM=
//  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=3400-4,VOL=SER=PARK2,DISP=(,PASS),
//      LABEL=(&FILE,NL,,IN),DCB=(RECFM=F,BLKSIZE=2048,DEN=2)
//SYSUT2 DD  DISP=SHR,DSN=MSS.S3026.P319(&MEM)
//SYSIN DD DUMMY
//   PEND
/*
//* * * * * * * * * * * * * * * * * * * * *
//   EXEC COPY,FILE=342,MEM=SEN1
//   EXEC COPY,FILE=343,MEM=SEN2
//   EXEC COPY,FILE=344,MEM=SEN3
//   EXEC COPY,FILE=345,MEM=SEN4
//   EXEC COPY,FILE=346,MEM=SEN5
//   EXEC COPY,FILE=347,MEM=SEN6
//   EXEC COPY,FILE=348,MEM=SEN7
//   EXEC COPY,FILE=349,MEM=SEN8
//   EXEC COPY,FILE=350,MEM=SEN9
//   EXEC COPY,FILE=351,MEM=SEN10
//   EXEC COPY,FILE=352,MEM=SEN11
//   EXEC COPY,FILE=353,MEM=SEN12
//   EXEC COPY,FILE=354,MEM=SEN13
//   EXEC COPY,FILE=355,MEM=SEN14
//   EXEC COPY,FILE=356,MEM=SEN15
//   EXEC COPY,FILE=357,MEM=SEN16
//   EXEC COPY,FILE=358,MEM=SEN17
//   EXEC COPY,FILE=359,MEM=SEN18
//   EXEC COPY,FILE=360,MEM=SEN19
/*
```

//

Transfer of the desired event data from MSS to the DISSPLA user's disk may now be performed. A batch fortran job with the appropriate FILEDEFs to denote the various geophone's data is submitted. The RSCS/NET feature is used to send the output of this routine to the user's reader. A sample fortran job for nine sensors follows:

```
//JLJ83026 JOB (3026,0304),'JOHNSTON',CLASS=A
//*MAIN ORG=NPGVM1.0090P
//  EXEC FORTXCG,REGION.GO=1024K
//FORT.SYSIN DD *
C
      LOGICAL*1 INFO1(8),INFO2(8),INFO3(8),INFO4(8),INFO5(8)
      LOGICAL*1 INFO6(8),INFO7(8),INFO8(8),INFO9(8)
      INTEGER*2 DATA1(1020),DATA2(1020),DATA3(1020),DATA4(1020)
      INTEGER*2 DATA5(1020),DATA6(1020),DATA7(1020),DATA8(1020)
      INTEGER*2 DATA9(1020),DATB1(4096)
C
      DO 30 J=1,4
         READ(1,100) INFO1,DATA1
         READ(2,100) INFO2,DATA2
         READ(3,100) INFO3,DATA3
         READ(4,100) INFO4,DATA4
         READ(8,100) INFO5,DATA5
         READ(9,100) INFO6,DATA6
         READ(10,100)INFO7, DATA7
         READ(11,100)INFO8, DATA8
         READ(12,100)INFO9, DATA9
100     FORMAT(8A1,102(10A2))
        DO 10 I = 10,1020,10
      WRITE(6,101) DATA1(I - 9),DATA1(I - 8),DATA1(I - 7),
     1DATA1(I - 6),DATA1(I - 5),DATA1(I - 4),DATA1(I - 3),
     2DATA1(I - 2),DATA1(I - 1),DATA1(I)
```

```
10     CONTINUE
       DO 20 I = 10,1020,10
       WRITE(6,101)DATA?(I - 9),DATA2(I - 8),DATA2(I - 7),
      1DATA2(I - 6),DATA2(I - 5),DATA2(I - 4),DATA2(I - 3),
      2DATA2(I - 2),DATA2(I - 1),DATA2(I)
20     CONTINUE
       DO 40 I = 10,1020,10
       WRITE(6,101)DATA3(I - 9),DATA3(I - 8),DATA3(I - 7),
      1DATA3(I - 6),DATA3(I - 5),DATA3(I - 4),DATA3(I - 3),
      2DATA3(I - 2),DATA3(I - 1),DATA3(I)
40     CONTINUE
       DO 50 I = 10,1020,10
       WRITE(6,101)DATA4(I - 9),DATA4(I - 8),DATA4(I - 7),
      1DATA4(I - 6),DATA4(I - 5),DATA4(I - 4),DATA4(I - 3),
      2DATA4(I - 2),DATA4(I - 1),DATA4(I)
50     CONTINUE
       DO 60 I = 10,1020,10
       WRITE(6,101)DATA5(I - 9),DATA5(I - 8),DATA5(I - 7),
      1DATA5(I - 6),DATA5(I - 5),DATA5(I - 4),DATA5(I - 3),
      2DATA5(I - 2),DATA5(I - 1),DATA5(I)
60     CONTINUE
       DO 70 I = 10,1020,10
       WRITE(6,101)DATA6(I - 9),DATA5(I - 8),DATA6(I - 7),
      1DATA6(I - 6),DATA6(I - 5),DATA6(I - 4),DATA6(I - 3),
      2DATA6(I - 2),DATA6(I - 1),DATA6(I)
70     CONTINUE
       DO 80 I = 10,1020,10
       WRITE(6,101)DATA7(I - 9),DATA7(I - 8),DATA7(I - 7),
      1DATA7(I - 6),DATA7(I - 5),DATA7(I - 4),DATA7(I - 3),
      2DATA7(I - 2),DATA7(I - 1),DATA7(I)
80     CONTINUE
       DO 90 I = 10,1020,10
       WRITE(6,101)DATA8(I - 9),DATA8(I - 8),DATA8(I - 7),
      1DATA8(I - 6),DATA8(I - 5),DATA8(I - 4),DATA8(I - 3),
      2DATA8(I - 2),DATA8(I - 1),DATA8(I)
```

```
90     CONTINUE
       DO 91 I =  10,1020,10
       WRITE(6,101)DATA9(I - 9),DATA9(I - 8),DATA9(I - 7),
      1DATA9(I - 6),DATA9(I - 5),DATA9(I - 4),DATA9(I - 3),
      2DATA9(I - 2),DATA9(I - 1),DATA9(I)
91     CONTINUE
101     FORMAT(I6,I6,I6,I6,I6,I6,I6,I6,I6,I6)
30      CONTINUE
        STOP
        END
/*
//GO.FT01F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN1)
//GO.FT02F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN2)
//GO.FT03F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN3)
//GO.FT04F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN4)
//GO.FT08F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN5)
//GO.FT09F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN6)
//GO.FT10F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN7)
//GO.FT11F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN8)
//GO.FT12F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN9)
//GO.SYSIN  DD  *
/*
//
```

A sample job for a six sensor group array follows:

```
//JLJ83026 JOB (3026,0304),'JOHNSTON',CLASS=A
//*MAIN ORG=NPGVM1.0090P
//  EXEC FORTXCG,REGION.GO=1024K
//FORT.SYSIN DD *
C
      LOGICAL*1 INFO1(8),INFO2(8),INFO3(8),INFO4(8),INFO5(8)
      LOGICAL*1 INFO6(8),INFO7(8),INFO8(8),INFO9(8)
      INTEGER*2 DATA1(1020),DATA2(1020),DATA3(1020),DATA4(1020)
      INTEGER*2 DATA5(1020),DATA6(1020),DATA7(1020),DATA8(1020)
      INTEGER*2 DATA9(1020),DATB1(4096)
```

```
C
      DO 30 J=1,4
         READ(1,100) INFO1,DATA1
         READ(2,100) INFO2,DATA2
         READ(3,100) INFO3,DATA3
         READ(4,100) INFO4,DATA4
         READ(8,100) INFO5,DATA5
         READ(9,100) INFO6,DATA6
100    FORMAT(8A1,102(10A2))
       DO 10 I = 10,1020,10
      WRITE(6,101)DATA1(I - 9),DATA1(I - 8),DATA1(I - 7),
     1DATA1(I - 6),DATA1(I - 5),DATA1(I - 4),DATA1(I - 3),
     2DATA1(I - 2),DATA1(I - 1),DATA1(I)
10    CONTINUE
      DO 20 I = 10,1020,10
      WRITE(6,101)DATA2(I - 9),DATA2(I - 8),DATA2(I - 7),
     1DATA2(I - 6),DATA2(I - 5),DATA2(I - 4),DATA2(I - 3),
     2DATA2(I - 2),DATA2(I - 1),DATA2(I)
20    CONTINUE
      DO 40 I = 10,1020,10
      WRITE(6,101)DATA3(I - 9),DATA3(I - 8),DATA3(I - 7),
     1DATA3(I - 6),DATA3(I - 5),DATA3(I - 4),DATA3(I - 3),
     2DATA3(I - 2),DATA3(I - 1),DATA3(I)
40    CONTINUE
      DO 50 I = 10,1020,10
      WRITE(6,101)DATA4(I - 9),DATA4(I - 8),DATA4(I - 7),
     1DATA4(I - 6),DATA4(I - 5),DATA4(I - 4),DATA4(I - 3),
     2DATA4(I - 2),DATA4(I - 1),DATA4(I)
50    CONTINUE
      DO 60 I = 10,1020,10
      WRITE(6,101)DATA5(I - 9),DATA5(I - 8),DATA5(I - 7),
     1DATA5(I - 6),DATA5(I - 5),DATA5(I - 4),DATA5(I - 3),
     2DATA5(I - 2),DATA5(I - 1),DATA5(I)
60    CONTINUE
```

```
      DO 70 I = 10,1020,10
      WRITE(6,101)DATA6(I - 9),DATA6(I - 8),DATA6(I - 7),
     1DATA6(I - 6),DATA6(I - 5),DATA6(I - 4),DATA6(I - 3),
     2DATA6(I - 2),DATA6(I - 1),DATA6(I)
70    CONTINUE
101     FORMAT(I6,I6,I6,I6,I6,I6,I6,I6,I6,I6)
30      CONTINUE
        STOP
        END
/*
//GO.FT01F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN3)
//GO.FT02F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN6)
//GO.FT03F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN9)
//GO.FT04F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN12)
//GO.FT08F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN15)
//GO.FT09F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN18)
//GO.SYSIN  DD  *
/*
//
```

Two files will be returned to user's reader. The first file is the listing and diagnostics file and should be purged. The second file should be named SEN DATA. SEN DATA must now be edited. Delete the first seven lines of the file and issue the command LREC 80 to set the proper file record length.

The interactive program can be run with the complete collection of files listed in the exec MATCH. The MATCH EXEC follows:

```
&TRACE OFF
FORTGI MFILTER
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLSP NONIMSL
FILEDEF 10 TERMINAL
FILEDEF 05 DISK SEN DATA (PERM)
FILEDEF 07 DISK COM DATA (PERM)
```

```
FI 4 DISK FILTER DATA (RECFM VS PERM
FILEDEF 18 DISK DISSPLA METAFILE T4 (RECFM VBS LRECL
                                        19065 BLOCK 19069
EXEC DISSPLA MFILTER
```

Entry parameters, such as the number of sensors in the array and the sampling rate, can be found in the NOSC data log for the event under study. All other interactive entries are user selected options or are self explanatory. [Ref. 12]

APPENDIX B

SAMPLE INTERACTIVE PROGRAM SESSION

```
match
G1 COMPILER ENTERED
SOURCE ANALYZED
PROGRAM NAME = MAIN
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = ANGLE
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = TIMOUT
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = FREQOT
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = MATCH
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = MYDATA
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = MAXMIN
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = SPCTRM
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = MULTI
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
```

```
PROGRAM NAME = MLTPLT
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = RMS
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = AVG
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = SIMULT
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = LMS
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = PLT
*  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = SOLV
*  NO DIAGNOSTICS GENERATED
    *STATISTICS*  NO DIAGNOSTICS THIS STEP
DISK 'T' NOT ACCESSED.
B (126) R/O
C (127) R/O
E (128) R/O

          ... Your Fortran program is now being loaded ...
                ... execution will soon follow ...
EXECUTION BEGINS...

ENTER EVENT RECORDING NUMBER-I3-
383

EVENT NUMBER 383

383. A33. 10 SEPT81 5KM EOD SHOT. SET B. NO DELAY. C141 ON FINAL
```

```
   AT END OF TAPE."
ENTER SAMPLE RATE IN HERTZ-REAL-
120.
ENTER LOW LOOK ANGLE IN DEGREES-I3-
-100
ENTER HIGH LOOK ANGLE IN DEGREES-I3-
200
ENTER MATCH FILTER THRESHOLD. RANGE OF 0. TO 1.0
.9
ENTER PLOT SCALING FACTOR-REAL-
.85
ENTER NUMBER OF SENSORS IN RING 6 OR 9 ONLY-I1-
9
ENTER SENSOR NUMBER FOR DISPLAY-I1-
4
ENTER NOISE THRESHOLD LEVEL -REAL-
1000.
ENTER DATA WINDOW SIZE FOR DIRECTION FINDING-I4-
0400
FOR COMPRS OUTPUT ENTER -1-. FOR TEK618 ENTER -2-
1
TO CREATE SIMULATED TARGETS ENTER -1- ELSE -2-
1
ENTER THE FOUR SIMULATION FREQUENCIES-REAL-
ENTER FREQUENCY
0
ENTER FREQUENCY
0
```

```
ENTER FREQUENCY
0
ENTER FREQUENCY
120.
ENTER AMPLITUDES FOR EACH FREQUENCY-REAL-
ENTER AMPLITUDE
0
ENTER AMPLITUDE
0
ENTER AMPLITUDE
0
ENTER AMPLITUDE
4000.
ENTER TARGET ANGLE FOR EACH FREQUENCY-I3-
SIX SENSORS ALLOWABLE ANGLES;0,60,120,180,240,300
FOR NINE SENSORS;0,40,80,120,160,200,240,280,320
ENTER ANGLE
0
ENTER ANGLE
0
ENTER ANGLE
0
ENTER ANGLE
120
TO MODIFY AMPLITUDE OF SIGNAL ABOVE NOISE THRESHOLD
ENTER - 1 -,ELSE ENTER - 2 -
```

```
2

ENTER DEGREE OF POLYNOMIAL DESIRED -I1-

FOR SIX SENSORS ENTER 2 - 4, FOR NINE ENTER 2 - 7
4
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.

CATALOG AS TARGET? IF YES TYPE - 1 -, ELSE - 2 -
2

FOR MATCH FILTER DIREC FINDING ENTER - 1 - ELSE -2-
1

FOR PHASE DELAY METHOD ENTER -1,FOR LMS ENTER -2
2

ENTER DEGREE OF POLYNOMIAL DESIRED -I1-

FOR SIX SENSORS ENTER 2 - 4, FOR NINE ENTER 2 - 7
4

TO VIEW OTHER SENSORS ENTER - 1 -
1

ENTER - 2 - TO CONTINUE TO NEXT TIME FRAME
2
```

APPENDIX C

PROGRAM LISTING

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C    PROGRAM MFILTER IS AN INTERACTIVE PROGRAM USED FOR EVALUATION OF  C
C    SEISMIC DATA.  DISCRETE MATCHED FILTERING IS USED TO CLASSIFY     C
C    TARGETS FROM THEIR SEISMIC SIGNATURES.  TARGET DIRECTIONS ARE     C
C    FOUND USING THE RELATIVE MATCHED FILTER PEAK POSITIONS.  THE      C
C    PROGRAM OUTPUTS FOR EACH 1024 SAMPLE PERIOD, THE AMPLITUDE        C
C    VERSUS TIME, POWER VERSUS FREQUENCY, MATCHED FILTER AMPLITUDE     C
C    VERSUS POSITION, AND THE TARGET CLASSES FOUND WITH THEIR RESPEC-  C
C    TIVE DIRECTIONS.  SIMULATION OF SEISMIC TARGETS IS SUPPORTED FOR  C
C    THE VALIDATION OF THE CLASSIFICATION AND DIRECTION ROUTINES.      C
C    SIMULATED TARGET SPECIFICATIONS ARE ENTERED INTERACTIVELY.        C
C    PROMPTING MESSAGES JUST PRIOR TO ALL USER ENTRIES, SPECIFY THE    C
C    TYPE OF ENTRY REQUIRED.  A PROMPT FOR A REAL NUMBER REQUIRES THAT C
C    A DECIMAL POINT BE INCLUDED IN THE ENTRY.  A PROMPT FOR AN INTEGERC
C    NUMBER SPECIFIES THE NUMBER OF DIGITS REQUIRED I.E.,IF -I4- IS    C
C    SPECIFIED AND THE NUMBER TWENTY IS TO BE ENTERED, "0020" MUST BE  C
C    ENTERED.                                                          C
C                                                                     C
C                          GLOBAL VARIABLES                           C
C                          ----------------                           C
C                                                                     C
C    VARIABLE    TYPE      DESCRIPTION                                C
C    NAME                                                             C
C                                                                     C
C    DATA1-9     INTEGER   ARRAYS OF EXPERIMENTAL DATA LOADED FROM A  C
C                          DATA FILE                                  C
C    PWR1-9      REAL      ARRAYS FOR SENSORS RELATIVE SPECTRAL POWERS C
C    IORR,II     INTEGER   SENSOR NUMBER FOR DISPLAY OR ANALYSIS      C
C    IWK         INTEGER   ARRAY USED BY THE FFT ROUTINE              C
C    DIRECT      INTEGER   VALUE OF INITIAL PRIMARY TARGET DIRECTION  C
C    DIRC        REAL      VALUE OF TARGET DIRECTION                  C
C    LOW,HIGH    INTEGER   LOW AND HIGH ANGLES SPECIFYING THE SECTOR  C
C                          FOR TARGETS OF INTEREST                    C
C    E           INTEGER   VALUE SELECTS COMPRS OR TEK618 OUTPUT      C
C                          FOR PLOTS                                  C
C    TESTNO      INTEGER   EVENT NUMBER UNDER ANALYSIS OR TO BE PLOTTEDC
C    COMMEN,     REAL*4    ARRAYS USED FOR INPUT AND OUTPUT OF        C
C    COMM1,                THE COMMENTS OF THE EXPERIMENTAL EVENT     C
C    COMM2                 UNDER EVALUATION                           C
C    SF          REAL      SAMPLING FREQUENCY                         C
C    F           REAL      FREQUENCY INCREMENT                        C
C    T           REAL      TIME INCREMENT                             C
C    XT,XF       REAL      ARRAYS USED IN THE CALCULATION OF THE TIME C
C                          AND FREQUENCY ARRAY RANGES                 C
C    TIME        REAL      TIME AXIS ARRAY                            C
C    FREQ        REAL      FREQUENCY AXIS ARRAY                       C
C    RA1-9,R1-9  REAL      ARRAYS OF EXPERIMENTAL DATA                C
```

```
C     B            REAL        ARRAY USED BY LIBRARY ROUTINE              C  MFI0C490
C     TH           REAL        ARRAY OF RELATIVE SENSOR ARRAY ANGLES      C  MFI0C500
C     RMAT         REAL        ARRAY USED FOR COMPUTATION AND STORAGE OF  C  MFI0C510
C                              MATCHED FILTER OUTPUT                      C  MFI0C520
C     UNIT         REAL        ARRAY USED FOR POSITION AXIS IN THE  MATCHEDC  MFI0C530
C                              FILTER PLOT                                C  MFI0C540
C     CAT,SCAT     REAL        ARRAY USED FOR STORAGE OF ONE SAMPLE SIGNAL C  MFI0C550
C     ECAT,TCAT    REAL        ARRAY CONTAINING ALL FILTER SAMPLE SIGNALS C  MFI0C560
C     RMAG,RA      REAL        ARRAY FOR PASSING ARRAYS RA1-9             C  MFI0C570
C     PWR          REAL        ARRAY FOR PASSING ARRAYS PWR1-9            C  MFI0C580
C     DIR          REAL        ARRAY OF THE DIRECTIONS OF THE TARGET CLASSES  MFI0C590
C                              FOUND BY THE MULTIPLE DIRECTION ROUTINE    C  MFI0C600
C     PLATE        REAL        ARRAY CONTAINING SEGMENTS OF RMAT          C  MFI0C610
C     THRES        REAL        VALUE IS THRESHOLD SPECIFIED FOR MATCHED   C  MFI0C620
C                              FILTER DETECTION OF THE FIVE TARGET CLASSES C  MFI0C630
C     NUMSEN       INTEGER     NUMBER OF SENSORS IN THE RING              C  MFI0C640
C     W,A,P        REAL        ARRAYS FOR THE FREQUENCY, AMPLITUDE, DIREC- C  MFI0C650
C                              TION AND PHASE OF THE SIMULATED TARGETS    C  MFI0C660
C     ND           INTEGER     ARRAY OF SIMULATED TARGET DIRECTIONS          MFI0C670
C     A1-9         COMPLEX     ARRAYS USED FOR FINDING AND STORING THE    C  MFI0C680
C                              FREQUENCY CONTENT OF ARRAYS RA1-9          C  MFI0C690
C     SCALE        REAL        SETS THE SIZE OF GRAPHIC OUTPUT            C  MFI0C700
C     TS,TF        REAL        FIRST AND LAST VALUE OF THE PLOTTING TIME  C  MFI0C710
C                              SCALE                                      C  MFI0C720
C     ITRGET       INTEGER     ARRAY OF SENSOR MATCHED FILTER PEAK        C  MFI0C730
C                              FOR EACH TARGET TYPE                       C  MFI0C740
C     NTYP         INTEGER     ARRAY OF PEAK POSITIONS FOR A TARGET       C  MFI0C750
C     JL           INTEGER     SAMPLE PERIOD NUMBER                       C  MFI0C760
C     NT,NW        INTEGER     NUMBER OF POINTS USED FOR MATCHED FILTERING C  MFI0C770
C     NFRM         INTEGER     TARGET TYPE NUMBER                         C  MFI0C780
C     WKAREA       REAL        ARRAY USED BY THE MATRIX SOLVING ROUTINE   C  MFI0C790
C     NOR          INTEGER     ORDER OF LEAST MEAN SQUARES POLYNOMIAL     C  MFI0C800
C                                                                         C  MFI0C810
C                                                                         C  MFI0C820
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC  MFI0C830
C                                                                         C  MFI0C840
      DIMENSION TRIX(9,9),B(9),WKAREA(108)                                   MFI0C850
      INTEGER*2 DATA1(1024),DATA2(1024),DATA3(1024),DATA4(1024),             MFI0C860
     1DATA5(1024),DATA6(1024),DATA7(1024),DATA8(1024),                       MFI0C870
     2DATA9(1024)                                                            MFI0C880
      INTEGER IWK(11),IORR,DIRECT,LOW,HIGH,C,TESTNC,                         MFI0C890
     1NTYP(6),ND(4),NUMSEN,NW,ITRGET(9,5),NTGT(9),NTAR(9)                    MFI0C900
      REAL*4 COMMEN(16),COMM1(16),COMM2(16)                                  MFI0C910
      REAL SF,F,T,PI,X1(1024),XF(1024),TIME(1024),FREQ(1024),                MFI0C920
     1RA1(1024),RA2(1024),RA3(1024),RA4(1024),RA5(1024),                     MFI0C930
     2RA6(1024),RA7(1024),RA8(1024),RA9(1024),DIR(5),DIRC,                   MFI0C940
     3PWR1(1024),PWR2(1024),PWR3(1024),PWR4(1024),PWR5(1024),                MFI0C950
     4PWR6(1024),PWR7(1024),PWR8(1024),PWR9(1024),                           MFI0C960
```

```
      5X(9),Y(360),W(9),TH(9),VX(360),TT(9),                            MF100970
      6RMAT(11264),UNIT(11264),CAT(1024),RMAG(1024),PWR(1024),          MF100980
      7RA(1024),TCEN(9),                                                MF100990
      8PLATE(1024),THRES,BCAT(5120),V(4),A(4),P(9,4)                    MF101000
       COMPLEX A1(1024),A2(1024),A3(1024),A4(1024),A5(1024),            MF101010
      2A6(1024),A7(1024),A8(1024),A9(1024)                              MF101020
C                                                                       MF101030
C                                                                       MF101040
C     INTERACTIVE INITIALIZATION OF PROGRAM                             MF101050
C                                                                       MF101060
      WRITE(6,79)                                                       MF101070
   79 FORMAT('0','ENTER EVENT RECORDING NUMBER-I3-')                    MF101080
      READ(10,80)TESTNO                                                 MF101090
   80 FORMAT(I3)                                                        MF101100
      WRITE(6,552)TESTNO                                                MF101110
  552 FORMAT('0','EVENT NUMBER ',I3)                                    MF101120
      DO 777 J = 1,60                                                   MF101130
         READ(7,559)NLEAD,COMMEN                                        MF101140
         READ(7,560)COMM1                                               MF101150
         READ(7,560)COMM2                                               MF101160
  560    FORMAT(16A4)                                                   MF101170
  559    FORMAT(3X,I3,16A4)                                             MF101180
         IF((NLEAD).EQ.(TESTNO)) WRITE(6,720)NLEAD,COMMEN               MF101190
  720    FORMAT('0',I3,16A4)                                            MF101200
         IF((NLEAD).EQ.(TESTNO)) WRITE(6,721)COMM1                      MF101210
         IF((NLEAD).EQ.(TESTNO)) WRITE(6,721)COMM2                      MF101220
  721    FORMAT('0',16A4)                                               MF101230
  777 CONTINUE                                                          MF101240
      WRITE (6,47)                                                      MF101250
   47 FORMAT('0','ENTER SAMPLE RATE IN HERTZ-REAL-')                    MF101260
      READ(10,78)SF                                                     MF101270
   78 FORMAT(F4.1)                                                      MF101280
      WRITE(6,81)                                                       MF101290
   81 FORMAT('0','ENTER LOW LOOK ANGLE IN DEGREES-I3-')                 MF101300
   83 FORMAT('0','ENTER HIGH LOOK ANGLE IN DEGREES-I3-')                MF101310
      READ(10,82)LOW                                                    MF101320
      WRITE(6,83)                                                       MF101330
      READ(10,82)HIGH                                                   MF101340
   82 FORMAT(I4)                                                        MF101350
      WRITE (6,19)                                                      MF101360
   19 FORMAT ('0','ENTER MATCH FILTER THRESHOLD. RANGE OF 0. TO 1.0')   MF101370
      READ (10,17) THRES                                                MF101380
   17 FORMAT (F3.2)                                                     MF101390
      WRITE (6,11)                                                      MF101400
   11 FORMAT('0','ENTER PLOT SCALING FACTOR-REAL-')                     MF101410
      READ(10,22)SCALE                                                  MF101420
   22 FORMAT(F4.2)                                                      MF101430
      WRITE(6,65)                                                       MF101440
```

```
85    FORMAT('0','ENTER NUMBER OF SENSORS IN RING 6 OR 9 ONLY-I1-')
      READ(10,15)NUMSEN
      WRITE(6,13)
13    FORMAT('0','ENTER SENSOR NUMBER FOR DISPLAY-I1-')
      READ (10,15)IORR
15    FORMAT(I1)
      WRITE(6,137)
137   FORMAT('0','ENTER NOISE THRESHOLD LEVEL -REAL-')
      READ(10,77)NOISE
77    FORMAT(F5.1)
      WRITE(6,201)
201   FORMAT('0','ENTER DATA WINDOW SIZE FOR DIRECTION FINDING-I4-')
      READ(6,62)NW
      IF((NW).GT.(1024)) NW = 1024
      WRITE(6,138)
138   FORMAT('0','FOR COMPRS OUTPUT ENTER -1-. FOR TEK618 ENTER -2-')
      READ(10,15)C
      WRITE(6,139)
139   FORMAT('0','TO CREATE SIMULATED TARGETS ENTER -1- ELSE -2-')
      READ(10,27)NSIMUL
27    FORMAT(I1)
      REDUCE = 1
      M=10
      N=1024
      N2=N/2
      DIRECT = 0
      F=SF/FLOAT(N)
      T=1./SF
      PI=3.141593
      DO 333 JIM =1,6
      NTYP(JIM) = 0
333   CONTINUE
C
C   LOAD FILTER SAMPLE SIGNAL ARRAY FROM DATA FILE
C
      DO 88 IJ = 4,5120,4
        READ(4,633) BCAT(IJ - 3),BCAT(IJ - 2),BCAT(IJ - 1),BCAT(IJ)
633   FORMAT(3X,F6.2,3X,F6.2,3X,F6.2,3X,F6.2)
88    CONTINUE
C
C   CALCULATE TIME AND FREQUENCY RANGES.
C   THE VALUE FOR -JL- ALSO DENOTES THE TIME PERIOD OF THE SENSOR DATA
C   E.G., JL = 1 MAY MEAN 0 - 8 SECONDS, JL = 2 MAY MEAN 8 - 16 SECONDS
C   ETC.
C
      DO 876 IQ = 1,4
      V(IQ) = 0.
      ND(IQ) = 0
```

```
      A(IC) = 0.                                                        MFI01930
876   CONTINUE                                                          MFI01940
      DO 10 JUMP = 1,4                                                  MFI01950
      JL = JUMP                                                         MFI01960
      DO 20 J=1,1024                                                    MFI01970
         XT(J)=FLOAT(J)                                                 MFI01980
         XF(J)=FLOAT(J)                                                 MFI01990
         TIME(J)=XT(J)*T + (JL - 1)*1024*T                              MFI02000
         FREQ(J)=XF(J)*F                                                MFI02010
20    CONTINUE                                                          MFI02020
C                                                                       MFI02030
C     LOAD THE SENSOR DATA ARRAYS WITH THEIR RESPECTIVE DATA FOR THE TIME MFI02040
C     PERIOD                                                            MFI02050
C                                                                       MFI02060
      DO 100 I = 10,1020,10                                             MFI02070
         READ(5,7)DATA1(I - 9),DATA1(I - 8),DATA1(I - 7),DATA1(I - 6),  MFI02080
     1DATA1(I - 5),DATA1(I - 4),DATA1(I - 3),DATA1(I - 2),              MFI02090
     2DATA1(I - 1),DATA1(I)                                             MFI02100
100   CONTINUE                                                          MFI02110
7     FORMAT(I6,I6,I6,I6,I6,I6,I6,I6,I6,I6)                             MFI02120
      DO 110 I = 10,1020,10                                             MFI02130
         READ(5,7)DATA2(I - 9),DATA2(I - 8),DATA2(I - 7),DATA2(I - 6),  MFI02140
     1DATA2(I - 5),DATA2(I - 4),DATA2(I - 3),DATA2(I - 2),              MFI02150
     2DATA2(I - 1),DATA2(I)                                             MFI02160
110   CONTINUE                                                          MFI02170
      DO 120 I = 10,1020,10                                             MFI02180
         READ(5,7)DATA3(I - 9),DATA3(I - 8),DATA3(I - 7),DATA3(I - 6),  MFI02190
     1DATA3(I - 5),DATA3(I - 4),DATA3(I - 3),DATA3(I - 2),              MFI02200
     2DATA3(I - 1),DATA3(I)                                             MFI02210
120   CONTINUE                                                          MFI02220
      DO 130 I = 10,1020,10                                             MFI02230
         READ(5,7)DATA4(I - 9),DATA4(I - 8),DATA4(I - 7),DATA4(I - 6),  MFI02240
     1DATA4(I - 5),DATA4(I - 4),DATA4(I - 3),DATA4(I - 2),              MFI02250
     2DATA4(I - 1),DATA4(I)                                             MFI02260
130   CONTINUE                                                          MFI02270
      DO 140 I = 10,1020,10                                             MFI02280
         READ(5,7)DATA5(I - 9),DATA5(I - 8),DATA5(I - 7),DATA5(I - 6),  MFI02290
     1DATA5(I - 5),DATA5(I - 4),DATA5(I - 3),DATA5(I - 2),              MFI02300
     2DATA5(I - 1),DATA5(I)                                             MFI02310
140   CONTINUE                                                          MFI02320
      DO 150 I = 10,1020,10                                             MFI02330
         READ(5,7)DATA6(I - 9),DATA6(I - 8),DATA6(I - 7),DATA6(I - 6),  MFI02340
     1DATA6(I - 5),DATA6(I - 4),DATA6(I - 3),DATA6(I - 2),              MFI02350
     2DATA6(I - 1),DATA6(I)                                             MFI02360
150   CONTINUE                                                          MFI02370
      IF((NUMSEN).EQ.(6)) GO TO 181                                     MFI02380
      DO 160 I = 10,1020,10                                             MFI02390
         READ(5,7)DATA7(I - 9),DATA7(I - 8),DATA7(I - 7),DATA7(I - 6),  MFI02400
```

```
     1DATA7(I - 5),DATA7(I - 4),DATA7(I - 3),DATA7(I - 2),           MFI02410
     2DATA7(I - 1),DATA7(I)                                            MFI02420
160   CONTINUE                                                          MFI02430
      DO 170 I = 10,1020,10                                             MFI02440
         READ(5,7)DATA8(I - 9),DATA8(I - 8),DATA8(I - 7),DATA8(I - 6), MFI02450
     1DATA8(I - 5),DATA8(I - 4),DATA8(I - 3),DATA8(I - 2),             MFI02460
     2DATA8(I - 1),DATA8(I)                                            MFI02470
170   CONTINUE                                                          MFI02480
      DO 180 I = 10,1020,10                                             MFI02490
         READ(5,7)DATA9(I - 9),DATA9(I - 8),DATA9(I - 7),DATA9(I - 6), MFI02500
     1DATA9(I - 5),DATA9(I - 4),DATA9(I - 3),DATA9(I - 2),             MFI02510
     2DATA9(I - 1),DATA9(I)                                            MFI02520
180   CONTINUE                                                          MFI02530
C                                                                       MFI02540
C    CONVERT THE INTEGER DATA TO REAL AND COMPLEX VALUES                MFI02550
C                                                                       MFI02560
181   CALL MYDATA(TIME,DATA1,N,RA1,A1)                                  MFI02570
      CALL MYDATA(TIME,DATA2,N,RA2,A2)                                  MFI02580
      CALL MYDATA(TIME,DATA3,N,RA3,A3)                                  MFI02590
      CALL MYDATA(TIME,DATA4,N,RA4,A4)                                  MFI02600
      CALL MYDATA(TIME,DATA5,N,RA5,A5)                                  MFI02610
      CALL MYDATA(TIME,DATA6,N,RA6,A6)                                  MFI02620
      IF((NUMSEN).EQ.(6)) GO TO 86                                      MFI02630
      CALL MYDATA(TIME,DATA7,N,RA7,A7)                                  MFI02640
      CALL MYDATA(TIME,DATA8,N,RA8,A8)                                  MFI02650
      CALL MYDATA(TIME,DATA9,N,RA9,A9)                                  MFI02660
C                                                                       MFI02670
C    IF SIMULATION IS REQUESTED, CALL THE SIMULATION ROUTINE            MFI02680
C                                                                       MFI02690
86    IF((NSIMUL).EQ.(1)) CALL SIMULT(RA1,RA2,RA3,RA4,RA5,RA6,          MFI02700
     1RA7,RA8,RA9,A1,A2,A3,A4,A5,A6,A7,A8,A9,V,A,ND,TNOISE,TIME,JL,P,  MFI02710
     2REDUCE,NUMSEN)                                                    MFI02720
C                                                                       MFI02730
C    CALCULATE THE FFT AND SPECTRAL FREQUENCY POWER FOR EACH SENSORS DATAMFI02740
C                                                                       MFI02750
      CALL FFT2C(A1,M,IWK)                                              MFI02760
      CALL SPCTRM(TIME,RA1,FREQ,A1,N,PWR1,PDB1,PH1)                     MFI02770
C                                                                       MFI02780
      CALL FFT2C(A2,M,IWK)                                              MFI02790
      CALL SPCTRM(TIME,RA2,FREQ,A2,N,PWR2,PDB2,PH2)                     MFI02800
C                                                                       MFI02810
      CALL FFT2C(A3,M,IWK)                                              MFI02820
      CALL SPCTRM(TIME,RA3,FREQ,A3,N,PWR3,PDB3,PH3)                     MFI02830
C                                                                       MFI02840
      CALL FFT2C(A4,M,IWK)                                              MFI02850
      CALL SPCTRM(TIME,RA4,FREQ,A4,N,PWR4,PDB4,PH4)                     MFI02860
C                                                                       MFI02870
      CALL FFT2C(A5,M,IWK)                                              MFI02880
```

```
      CALL SPCTRM(TIME,RA5,FREQ,A5,N,PWR5,PDE5,PH5)
C
      CALL FFT2C(A6,M,IWK)
      CALL SPCTRM(TIME,RA6,FREQ,A6,N,PWR6,PDE6,PH6)
C
      IF((NUMSEN).EQ.(6)) GO TO 87
      CALL FFT2C(A7,M,IWK)
      CALL SPCTRM(TIME,RA7,FREQ,A7,N,PWR7,PDE7,PH7)
C
      CALL FFT2C(A8,M,IWK)
      CALL SPCTRM(TIME,RA8,FREQ,A8,N,PWR8,PDE8,PH8)
C
      CALL FFT2C(A9,M,IWK)
      CALL SPCTRM(TIME,RA9,FREQ,A9,N,PWR9,PDE9,PH9)
C
C    DETERMINE IF A TARGET IS PRESENT AND IF THERE IS, IS IT IN THE
C    SECTOR OF INTEREST.
C
87    CALL MAXMIN(RA1,1024,K,RAGMAX,L,RIN)
      IF((ABS(RAGMAX)).GT.(TNOISE)) CALL ANGLE(DIRC,RA1,RA2,RA3,RA4,
     1RA5,RA6,RA7,RA8,RA9,NUMSEN,TESTNO,TIME,C,SCALE,NTAR)
      IF(((LOW).GT.(DIRECT)).OR.((HIGH).LT.(DIRECT))) GO TO 10
      CALL LMS (NTAR,DIRC,NUMSEN,C,SCALE,TESTNO,TIME,O,Y,VX,X,TT,INDEX,
     1NOR,TRIX,B,WKAREA)
      NFRM = C
      IF((NTAR(1)).GT.(0))CALL PLT(C,SCALE,TESTNO,NUMSEN,TIME,
     1Y,VX,X,TT,INDEX,NOR,DIRC,NFRM)
      DIRECT = IFIX(DIRC)
C
C    FOR THE SELECTED SENSOR IF A TARGET IS PRESENT, OUTPUT THE MATCHED
C    FILTER RESPONSE, THE AMPLITUDE VERSUS TIME  AND THE
C    SPECTRAL POWER VERSUS FREQUENCY PLOTS. IF SELECTED, OUTPUT THE
C    MULTIPLE TARGET INFORMATION.
C
99     IF((IORR).NE.(1)) GO TO 444
      CALL MAXMIN(RA1,1024,K,RAGMAX,L,RIN)
      IF((ABS(RAGMAX)).GT.(TNOISE)) GO TO 111
      WRITE(6,112)
      GO TO 444
112   FORMAT('0','NO DETECTABLE TARGET')
111   CALL MATCH(RA1,NTYP,SCALE,IORR,DIRECT,THRES,BCAT,TESTNO,JL,C,N,
     1TS,TF,TIME)
      CALL TIMOUT(TIME,RA1,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)
      CALL FREQOT(FREQ,PWR1,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)
C
C
C
C
444   IF((IORR).NE.(2)) GO TO 445
```

```
      CALL MAXMIN(RA2,1024,K,RAGMAX,L,RIN)
      IF((ABS(RAGMAX)).GT.(TNOISE)) GO TO 113
      WRITE(6,112)
      GO TO 445
113   CALL MATCH(RA2,NTYP,SCALE,ICRR,DIRECT,THRES,BCAT,TESTNO,JL,C,N,
     1TS,TF,TIME)
      CALL TIMOUT(TIME,RA2,DIRECT,SCALE,ICRR,NTYP,TESTNO,TS,TF,C)
      CALL FREQOT(FREQ,PWR2,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)
C
C
C
445   IF((IORR).NE.(3)) GO TO 446
      CALL MAXMIN(RA3,1024,K,RAGMAX,L,RIN)
      IF((ABS(RAGMAX)).GT.(TNOISE)) GO TO 114
      WRITE(6,112)
      GO TO 446
114   CALL MATCH(RA3,NTYP,SCALE,IORR,DIRECT,THRES,BCAT,TESTNO,JL,C,N,
     1TS,TF,TIME)
      CALL TIMOUT(TIME,RA3,DIRECT,SCALE,ICRR,NTYP,TESTNO,TS,TF,C)
      CALL FREQOT(FREQ,PWR3,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)
C
C
C
446   IF((IORR).NE.(4)) GO TO 447
      CALL MAXMIN(RA4,1024,K,RAGMAX,L,RIN)
      IF((ABS(RAGMAX)).GT.(TNOISE)) GO TO 115
      WRITE(6,112)
      GO TO 447
115   CALL MATCH(RA4,NTYP,SCALE,ICRR,DIRECT,THRES,BCAT,TESTNO,JL,C,N,
     1TS,TF,TIME)
      CALL TIMOUT(TIME,RA4,DIRECT,SCALE,ICRR,NTYP,TESTNO,TS,TF,C)
      CALL FREQOT(FREQ,PWR4,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)
C
C
C
447   IF((IORR).NE.(5)) GO TO 448
      CALL MAXMIN(RA5,1024,K,RAGMAX,L,RIN)
      IF((ABS(RAGMAX)).GT.(TNOISE)) GO TO 116
      WRITE(6,112)
      GO TO 448
116   CALL MATCH(RA5,NTYP,SCALE,IORR,DIRECT,THRES,BCAT,TESTNO,JL,C,N,
     1TS,TF,TIME)
      CALL TIMOUT(TIME,RA5,DIRECT,SCALE,ICRR,NTYP,TESTNO,TS,TF,C)
      CALL FREQOT(FREQ,PWR5,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)
C
C
C
448   IF((IORR).NE.(6)) GO TO 449
```

```
      CALL MAXMIN(RA6,1024,K,RAGMAX,L,RIN)                              MFI03850
      IF((ABS(RAGMAX)).GT.(TNOISE)) GO TO 117                           MFI03860
      WRITE(6,112)                                                      MFI03870
      GO TO 449                                                         MFI03880
117   CALL MATCH(RA6,NTYP,SCALE,IORR,DIRECT,THRES,BCAT,TESTNO,JL,C,N,   MFI03890
     1TS,TF,TIME)                                                       MFI03900
      CALL TIMOUT(TIME,RA6,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)       MFI03910
      CALL FRECOT(FREQ,PWR6,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)      MFI03920
C                                                                       MFI03930
C                                                                       MFI03940
C                                                                       MFI03950
449   IF((IORR).NE.(7)) GO TO 450                                       MFI03960
      CALL MAXMIN(RA7,1024,K,RAGMAX,L,RIN)                              MFI03970
      IF((ABS(RAGMAX)).GT.(TNOISE)) GO TO 118                           MFI03980
      WRITE(6,112)                                                      MFI03990
      GO TO 450                                                         MFI04000
118   CALL MATCH(RA7,NTYP,SCALE,IORR,DIRECT,THRES,BCAT,TESTNO,JL,C,N,   MFI04010
     1TS,TF,TIME)                                                       MFI04020
      CALL TIMOUT(TIME,RA7,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)       MFI04030
      CALL FRECOT(FREQ,PWR7,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)      MFI04040
C                                                                       MFI04050
C                                                                       MFI04060
C                                                                       MFI04070
450   IF((IORR).NE.(8)) GO TO 451                                       MFI04080
      CALL MAXMIN(RA8,1024,K,RAGMAX,L,RIN)                              MFI04090
      IF((ABS(RAGMAX)).GT.(TNOISE)) GO TO 119                           MFI04100
      WRITE(6,112)                                                      MFI04110
      GO TO 451                                                         MFI04120
119   CALL MATCH(RA8,NTYP,SCALE,IORR,DIRECT,THRES,BCAT,TESTNO,JL,C,N,   MFI04130
     1TS,TF,TIME)                                                       MFI04140
      CALL TIMOUT(TIME,RA8,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)       MFI04150
      CALL FRECOT(FREQ,PWR8,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)      MFI04160
C                                                                       MFI04170
C                                                                       MFI04180
C                                                                       MFI04190
451   IF((IORR).NE.(9)) GO TO 452                                       MFI04200
      CALL MAXMIN(RA9,1024,K,RAGMAX,L,RIN)                              MFI04210
      IF((ABS(RAGMAX)).GT.(TNOISE)) GO TO 125                           MFI04220
      WRITE(6,112)                                                      MFI04230
      GO TO 452                                                         MFI04240
125   CALL MATCH(RA9,NTYP,SCALE,IORR,DIRECT,THRES,BCAT,TESTNO,JL,C,N,   MFI04250
     1TS,TF,TIME)                                                       MFI04260
      CALL TIMOUT(TIME,RA9,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)       MFI04270
      CALL FRECOT(FREQ,PWR9,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,C)      MFI04280
C                                                                       MFI04290
C                                                                       MFI04300
C                                                                       MFI04310
452    IF((ABS(RAGMAX)).LT.(TNOISE)) GO TO 454                          MFI04320
```

```
      WRITE(6,837)                                                        MFI04330
837   FORMAT('0','FOR MATCH FILTER DIREC FINDING ENTER - 1 - ELSE -2-')   MFI04340
      READ(10,15) MNY                                                     MFI04350
      IF((MNY).EQ.(1))CALL MULTI(RA1,RA2,RA3,RA4,RA5,RA6,RA7,RA8,         MFI04360
     1RA9,NTYP,SCALE,IORR,THRES,BCAT,TESTNO,JL,C,TS,TF,V,A,ND,            MFI04370
     2NW,NUMSEN,TIME,ITRGET,DIRECT,DIR,DIRC)                              MFI04380
      IF((MNY).NE.(1)) GO TO 454                                          MFI04390
      WRITE(6,44)                                                         MFI04400
44    FORMAT('0','FOR PHASE DELAY METHOD ENTER -1,FOR LMS ENTER -2')      MFI04410
      READ(10,393)NGL                                                     MFI04420
393   FORMAT(I1)                                                          MFI04430
      DO 666 I21 = 1,9                                                    MFI04440
         NTAR(I21) = 0                                                    MFI04450
666   CONTINUE                                                            MFI04460
      IF((NGL).EQ.(1))GO TO 101                                           MFI04470
      DO 678 IW = 1,5                                                     MFI04480
       NTAR(1) = ITRGET(1,IW)                                             MFI04490
       NTAR(2) = ITRGET(2,IW)                                             MFI04500
       NTAR(3) = ITRGET(3,IW)                                             MFI04510
       NTAR(4) = ITRGET(4,IW)                                             MFI04520
       NTAR(5) = ITRGET(5,IW)                                             MFI04530
       NTAR(6) = ITRGET(6,IW)                                             MFI04540
       IF(NUMSEN.EQ.6) GO TO 646                                          MFI04550
       NTAR(7) = ITRGET(7,IW)                                             MFI04560
       NTAR(8) = ITRGET(8,IW)                                             MFI04570
       NTAR(9) = ITRGET(9,IW)                                             MFI04580
646    NG = IW                                                            MFI04590
       IF((NTAR(1)).EQ.(0)) GO TO 678                                     MFI04600
       CALL LMS (NTAR,DIRC,NUMSEN,C,SCALE,TESTNO,TIME,NG,Y,               MFI04610
     1VX,X,TT,INDEX,NOR,TRIX,B,WKAREA)                                    MFI04620
       CALL PLT(C,SCALE,TESTNO,NUMSEN,TIME,Y,VX,X,TT,INDEX,NOR,DIRC,      MFI04630
     1NG)                                                                 MFI04640
         DIR(NG) = DIRC                                                   MFI04650
678      CONTINUE                                                         MFI04660
       GO TO 601                                                          MFI04670
C                                                                         MFI04680
C    PHASE DELAY METHOD                                                   MFI04690
C                                                                         MFI04700
101   DO 60  J3 = 1,5                                                     MFI04710
         TC = 0                                                           MFI04720
         DO 70 K3 = 1,NUMSEN                                              MFI04730
           TCEN(K3) = FLOAT(ITRGET(K3,J3))                                MFI04740
           TC = FLOAT(ITRGET(K3,J3)) + TC                                 MFI04750
70       CONTINUE                                                         MFI04760
         TC = TC/FLOAT(NUMSEN)                                            MFI04770
         CB = 0.                                                          MFI04780
         DIVIDE = 0.                                                      MFI04790
         DO 840 N = 1,NUMSEN                                              MFI04800
```

```
      DO 90 M = 1,NUMSEN
         IF((M).LE.(N)) GO TO 90
         BB = ((TC - TCEN(M))*X(N) - (TC - TCEN(N))*X(M))
         AA= (TC - TCEN(N))*Y(M) - (TC - TCEN(M))*Y(N)
         IF((AA).EQ.(0)) GO TO 90
         IF((BB).LT.(.000001)) GO TO 90
         DIVIDE = DIVIDE + 1.
         QB = ATAN(BB/AA) + QB
90      CONTINUE
840    CONTINUE
   NUM = NUMSEN - 1
   QB = (QB/(DIVIDE))*(180.0/3.141593)
   DIR(J3) = QB
60 CONTINUE
C
601 CALL MLTPLT(C,SCALE,TESTNO,TS,TF,DIR,V,A,ND,ITRGET)
C
C   SELECT TO EVALUATE OTHER SENSORS IN THIS TIME PERIOD OR CONTINUE
C   TO THE NEXT TIME PERIOD
C
454 WRITE(6,51)
    WRITE(6,52)
51  FORMAT('0','TO VIEW OTHER SENSORS ENTER - 1 - ')
52  FORMAT('0','ENTER - 2 - TO CONTINUE TO NEXT TIME FRAME')
    READ(10,551)NS
551 FORMAT(I1)
    IF((NS).NE.(1)) GO TO 10
    WRITE(6,57)
57  FORMAT('0','ENTER NEW SENSOR NUMBER FOR DISPLAY-I1-')
    READ (10,551)ICRR
    GO TO 99
10   CONTINUE
    CALL DONEPL
    STOP
    END
C
C   SUBROUTINE ANGLE COMPUTES A ROUGH DIRECTION TO THE PRIMARY TARGET
C   FOR USE IN DETERMINING WHETHER THE TARGET IS IN
C   THE SECTOR OF INTEREST
C   TIME DOMAIN ANALYSIS IS USED.
C
      SUBROUTINE ANGLE (DIRC,R1,R2,R3,R4,R5,R6,R7,R8,R9,NUMSEN,TESTNO,
     1TIME,C,SCALE,NTAR)
      INTEGER M(9),TESTNO,NTAR(9),NUMSEN,C
      REAL R1(1024),R2(1024),R3(1024),R4(1024),R5(1024),R6(1024),
     1R7(1024),R8(1024),R9(1024),TIME(1024),DIR(5),DIRC
      DO 12 I = 1,9
         NTAR(I) = 0
```

```
12    CONTINUE
      CALL MAXMIN(R1,1024,M(1),RMAX,L,RMIN)
      CALL MAXMIN(R2,1024,M(2),RMAX,L,RMIN)
      CALL MAXMIN(R3,1024,M(3),RMAX,L,RMIN)
      CALL MAXMIN(R4,1024,M(4),RMAX,L,RMIN)
      CALL MAXMIN(R5,1024,M(5),RMAX,L,RMIN)
      CALL MAXMIN(R6,1024,M(6),RMAX,L,RMIN)
      IF((NUMSEN).EQ.(6)) GO TO 10
      CALL MAXMIN(R7,1024,M(7),RMAX,L,RMIN)
      CALL MAXMIN(R8,1024,M(8),RMAX,L,RMIN)
      CALL MAXMIN(R9,1024,M(9),RMAX,L,RMIN)
10    DO 11 I2 = 1,NUMSEN
        NTAR(I2) = M(I2)
11    CONTINUE
      RETURN
      END
C
C    SUBROUTINE TIMOUT PLOTS AMPLITUDE VERSUS TIME FOR THE SELECTED
C    SENSOR
C
      SUBROUTINE TIMOUT (TIME,RA,DIRECT,SCALE,IORR,NTYP,TESTNO,TS,TF,
     1C)
      REAL RA(1024),TIME(1024)
      INTEGER NTYP(6),C,NY,DIRECT,TESTNO
      IF((C).EQ.(1)) CALL COMPRS
      IF((C).NE.(1)) CALL TEK618
      CALL PAGE(11.0,8.5)
      CALL NOBRDR
      CALL BLOWUP(SCALE)
      CALL AREA2D(9.0,6.0)
      CALL FRAME
      CALL XNAME('TIME -SECONDS-',14)
      CALL YNAME('INPUT',5)
      CALL HEADIN('SENSOR INPUT - VS - TIME',24,2.0,1)
      CALL MESSAG('SENSOR NUMBER ',14,2.0,5.5)
      CALL INTNO(IORR,4.0,5.5)
      CALL MESSAG('EVENT NUMBER',12,2.0,5.7)
      CALL INTNO(TESTNO,4.0,5.7)
      CALL MESSAG('TARGET I.D.',11,2.0,5.3)
      XPOS = 2.80
      IF((NTYP(1)).NE.(0)) XPOS = XPOS + 1.2
      IF((NTYP(1)).NE.(0))
     1CALL MESSAG('TRKD-VEH',8,XPOS,5.3)
      IF((NTYP(2)).NE.(0)) XPOS = XPOS + 1.2
      IF((NTYP(2)).NE.(0))
     1CALL MESSAG('WHLD-VEH',8,XPOS,5.3)
      IF((NTYP(3)).NE.(0)) XPOS = XPOS + 1.2
      IF((NTYP(3)).NE.(0))
```

```
     1CALL MESSAG('SHL-BLST',8,XPOS,5.3)
      IF((NTYP(4)).NE.(0)) XPOS = XPOS + 1.2
      IF((NTYP(4)).NE.(0))
     1CALL MESSAG('HELICPTR',8,XPOS,5.3)
      IF((NTYP(5)).NE.(0)) XPOS = XPOS + 1.2
      IF((NTYP(5)).NE.(0))
     1CALL MESSAG('PERSONNEL',9,XPOS,5.3)
      CALL MESSAG('INIT DIRECTION',14,2.0,5.1)
      CALL INTNO(DIRECT,4.0,5.1)
      CALL INTAXS
      CALL YAXANG (0.)
      CALL XTICKS(10)
      CALL YTICKS(10)
      TS = FLOAT(IFIX(TIME(1)))
      TF = FLOAT(IFIX(TIME(1024) + 1.))
      CALL MAXMIN(RA,1024,K,RAMAX,L,RAMIN)
      IAMP = 1
      DO 337 NI = 100,5000,100
        IF((IAMP).NE.(1)) GO TO 337
        IF((NI).GE.(ABS(RAMAX))) IAMP = NI
337   CONTINUE
      STEP = IAMP/5.0
      CALL GRAF(TS,1.,TF,FLOAT(-IAMP),STEP,FLOAT(IAMP))
      CALL CURVE(TIME,RA,1024,0.)
      CALL ENDPL(0)
      WRITE(6,70)
70    FORMAT('0','CATALOG AS TARGET? IF YES TYPE - 1 -, ELSE - 2 -')
      READ(10,71)NY
71    FORMAT(I1)
C
C     SEISMIC DATA DESIRED TO BE USED AS A SAMPLE SIGNAL IN THE MATCHED
C     FILTER CAN BE STORED IN THE FILTER DATA FILE.
C
      IF((NY).NE.(1)) GO TO 45
      DO 44 JJ = 4,1024,4
      IF((ABS(RA(JJ))).LT.(.001)) RA(JJ) = 0
      IF((ABS(RA(JJ - 1))).LT.(.001)) RA(JJ - 1) = 0
      IF((ABS(RA(JJ -2))).LT.(.001)) RA(JJ - 2) = 0
      IF((ABS(RA(JJ -3))).LT.(.001)) RA(JJ - 3) = 0
      WRITE (4,3)RA(JJ -3),RA(JJ -2),RA(JJ -1),RA(JJ)
3     FORMAT(3X,F9.2,3X,F9.2,3X,F9.2,3X,F9.2)
44    CONTINUE
45    RETURN
      END
C
C     SUBROUTINE FREQOT PLOTS SPECTRAL POWER VERSUS FREQUENCY
C
      SUBROUTINE FREQOT(FREQ,PWR,DIRECT,SCALE,IORF,NTYP,TESTNC,TS,TF,
```

```
1C)                                                                 MFI06250
 REAL PWR(1024),FREQ(1024)                                          MFI06260
 INTEGER NTYP(6),C,DIRECT,TESTNO                                    MFI06270
 IF((C).EQ.(1)) CALL CCMPRS                                         MFI06280
 IF((C).NE.(1)) CALL TEK618                                         MFI06290
 CALL PAGE(11.0,8.5)                                                MFI06300
 CALL NCBRDR                                                        MFI06310
 CALL BLOWUP(SCALE)                                                 MFI06320
 CALL AREA2D(9.0,6.0)                                               MFI06330
 CALL FRAME                                                         MFI06340
 CALL XNAME('FREQUENCY -HZ-',14)                                    MFI06350
 CALL YNAME('POWER',5)                                              MFI06360
 CALL HEADIN('SENSOR POWER -VS- FREQUENCY',27,2.0,1)                MFI06370
 CALL MESSAG('EVENT NUMBER ',13,2.0,5.7)                            MFI06380
 CALL INTNC(TESTNO,4.0,5.7)                                         MFI06390
 CALL MESSAG('SENSOR NUMBER ',14,2.0,5.5)                           MFI06400
 CALL INTNC(IORR,4.0,5.5)                                           MFI06410
 CALL MESSAG('TARGET I.D.',11,2.0,5.3)                              MFI06420
 XPOS = 2.80                                                        MFI06430
 IF((NTYP(1)).NE.(0)) XPOS = XPOS + 1.2                             MFI06440
 IF((NTYP(1)).NE.(0))                                               MFI06450
1CALL MESSAG('TRKD-VEH',8,XPOS,5.3)                                 MFI06460
 IF((NTYP(2)).NE.(0)) XPOS = XPOS + 1.2                             MFI06470
 IF((NTYP(2)).NE.(0))                                               MFI06480
1CALL MESSAG('WHLD-VEH',8,XPOS,5.3)                                 MFI06490
 IF((NTYP(3)).NE.(0)) XPOS = XPOS + 1.2                             MFI06500
 IF((NTYP(3)).NE.(0))                                               MFI06510
1CALL MESSAG('SHL-BLST',8,XPOS,5.3)                                 MFI06520
 IF((NTYP(4)).NE.(0)) XPOS = XPOS + 1.2                             MFI06530
 IF((NTYP(4)).NE.(0))                                               MFI06540
1CALL MESSAG('HELICPTR',8,XPOS,5.3)                                 MFI06550
 IF((NTYP(5)).NE.(0)) XPOS = XPOS + 1.2                             MFI06560
 IF((NTYP(5)).NE.(0))                                               MFI06570
1CALL MESSAG('PERSONNEL',9,XPOS,5.3)                                MFI06580
 CALL MESSAG('INIT DIRECTION',14,2.0,5.1)                           MFI06590
 CALL INTNC(DIRECT,4.0,5.1)                                         MFI06600
 CALL MESSAG('TIME PERIOD(SEC)',16,2.0,4.9)                         MFI06610
 CALL REALNC(TS,2,3.9,4.9)                                          MFI06620
 CALL REALNO(TF,2,4.7,4.9)                                          MFI06630
 XMAX = 1.0                                                         MFI06640
 DO 39 I = 20,500,20                                                MFI06650
    IF((XMAX).NE.(1.0)) GO TO 39                                    MFI06660
    IF((I).GE.(IFIX(FREQ(512)))) XMAX = FLOAT(I)                    MFI06670
39 CONTINUE                                                         MFI06680
 XSTEP = XMAX/4.00                                                  MFI06690
 CALL MAXMIN(PWR,1024,KP,PWRMAX,LP,PWRMIN)                          MFI06700
 NS = IFIX(XSTEP)                                                   MFI06710
 IF((XSTEP).GT.(20.0)) NS = IFIX(FLOAT(NS)/10.0)                    MFI06720
```

```
      CALL INTAXS
      CALL YAXANG (0.)
      CALL XTICKS(NS)
      CALL YTICKS(10)
      CALL GRAF(0.,XSTEP,XMAX,0.,.1,PWRMAX)
      CALL CURVE(FREQ,PWR,512,0.)
      CALL RESET ('ALL')
      CALL ENDPL(0)
38    CONTINUE
37    RETURN
      END
C
C    SUBROUTINE MATCH PERFORMS THE MATCHED FILTERING OF THE SELECTED
C    SENSORS DATA.  TARGET CLASSIFICATIONS FOUND ARE RETURNED, AS ARE
C    THEIR RESPECTIVE PEAK SIGNAL DETECTION POSITIONS.
C
      SUBROUTINE MATCH(RMAG,NTYP,SCALE,ICRR,DIRECT,THRES,TCAT,TESTNO,
     1JL,C,NT,TS,TF,TIME)
      REAL RMAG(NT),CAT(1024),UNIT(11264),PLATE(2048),SCAT(1024),
     1RMAT(11264),THRES,XRAY4(2),YRAY(2),TIME(1024),
     2XRAY1(2),XRAY2(2),XRAY3(2),TCAT(5120)
      INTEGER NTYP(6),C,DIRECT,TESTNO
C
      DATA XRAY1/2048.,2048./
      DATA XRAY2/4096.,4096./
      DATA XRAY3/6144.,6144./
      DATA XRAY4/8192.,8192./
      DATA YRAY/-1.0,1.0/
      DO 33 J = 1,1024
        PLATE(J) = 0.
        PLATE(J + 1024) = 0.
        SCAT(J) = 0.
        CAT(J) = 0.
33    CONTINUE
      NH = NT/2
      N2 = NT*2
      N4 = NT*4
      N6 = NT*6
      N8 = NT*8
      N10 = NT*10
      N11 = NT*11
      DO 600 MM = 1,N11
      RMAT(MM) = 0.0
      UNIT(MM) = MM
600   CONTINUE
      DO 201 K = N2,N10,N2
      NM = K - NT
       DO 202 KK = 1,NT
```

```
      RMAT(NM + KK) = RMAG(KK)                                          MFI07210
202   CONTINUE                                                          MFI07220
201   CONTINUE                                                          MFI07230
      NTYP(1) = 0                                                       MFI07240
      NTYP(2) = 0                                                       MFI07250
      NTYP(3) = 0                                                       MFI07260
      NTYP(4) = 0                                                       MFI07270
      NTYP(5) = 0                                                       MFI07280
C                                                                       MFI07290
C    NEW FILTER SAMPLE SIGNALS ARE LOADED AFTER EACH SUCCESSIVE 2048    MFI07300
C    ELEMENTS OF THE WORKING ARRAY 'RMAT' ARE CONVOLVED                 MFI07310
C                                                                       MFI07320
111   DO 400 I =1,N10                                                   MFI07330
      IF(((I).EQ.(1)).OR.((I).EQ.(N2)).OR.((I).EQ.(N4)).OR.             MFI07340
     1((I).EQ.(N6)).OR.((I).EQ.(N8)))                                   MFI07350
     2GO TO 98                                                          MFI07360
      GO TO 99                                                          MFI07370
98    IF((I).EQ.(1)) MARK = 1025                                        MFI07380
      IF((I).EQ.(N2)) MARK = 2049                                       MFI07390
      IF((I).EQ.(N4)) MARK = 3073                                       MFI07400
      IF((I).EQ.(N6)) MARK = 4097                                       MFI07410
      IF((I).EQ.(N8)) MARK = 5121                                       MFI07420
C                                                                       MFI07430
C    FILTER DATA REVERSED                                               MFI07440
C                                                                       MFI07450
      DO 777 LL = 1,NT                                                  MFI07460
      CAT(LL) = TCAT(-LL + MARK)                                        MFI07470
777   CONTINUE                                                          MFI07480
C                                                                       MFI07490
C    EXECUTION TIME MAY BE REDUCED BY SELECTING SMALLER WINDOWS OR      MFI07500
C    SEGMENTS AROUND THE PEAK AMPLITUDES OF THE FILTER AND SEISMIC DATA MFI07510
C                                                                       MFI07520
      CALL MAXMIN(CAT,1024,LL,CATM,KK,CATMIN)                           MFI07530
      DO 10 II = 1,NT                                                   MFI07540
        IF((LL - NH).LE.(0)) LL = NH                                    MFI07550
        IF((LL).GT.(1024 - NH)) LL = 1024 - NH                          MFI07560
        SCAT(II) = CAT(II - NH + LL)                                    MFI07570
10     CONTINUE                                                         MFI07580
99    CALL RMS(SCAT,NT,CATMAX)                                          MFI07590
      CALL RMS(RMAG,NT,RAGMAX)                                          MFI07600
      CALL AVG(SCAT,NT,ACAT)                                            MFI07610
      RMAT(I) = 0                                                       MFI07620
C                                                                       MFI07630
C   REMOVAL OF DC BIAS AND NORMALIZATION OF THE FILTER AND SEISMIC DATA MFI07640
C                                                                       MFI07650
        DO 200 KN = 1,NT                                                MFI07660
        IF(((CATMAX).EQ.(0)).OR.((RAGMAX).EQ.(0))) GO TO 200            MFI07670
C                                                                       MFI07680
```

```
C     DISCRETE CONVOLUTION PREFORMED
C
        RMAT(I) = ((RMAT(I + KN))/(ABS(RAGMAX)))*
       1((SCAT(KN) - ACAT)/(CATMAX)) + RMAT(I)
  200   CONTINUE
  400   CONTINUE
        CALL MAXMIN(RMAT,N11,LZ,RATMAX,KZ,RATMIN)
        DO 390 NN = 1,N11
        IF((RATMAX).EQ.(0)) GO TO 390
        RMAT(NN) = RMAT(NN)/(ABS(RATMAX))
  390   CONTINUE
C
C     CLASSIFICATION OF TARGETS BY TARGET TYPE
C
        DO 302 J1 = N2,N10,N2
           DO 301 I1 = 1,N2
              PLATE(I1) = RMAT(J1 - N2 + I1)
  301      CONTINUE
           CALL MAXMIN(PLATE,N2,LP,PMAX,LS,PMIN)
           IF((ABS(PMAX)).GE.(THRES)) NTYP(J1/N2) = LP
  302   CONTINUE
C
C     MATCHED FILTER OUTPUT PLOTTING
C
        IF((IORR).EQ.(0)) GO TO 391
        IF((C).EQ.(1)) CALL COMPRS
        IF((C).NE.(1)) CALL TEK618
        CALL PAGE(11.0,8.5)
        CALL NOBRDR
        CALL BLOWUP(SCALE)
        CALL AREA2D(9.0,6.0)
        CALL FRAME
        CALL XNAME('POSITION',8)
        CALL YNAME('AMPLITUDE OF MATCHED FILTER',27)
        CALL HEADIN('MATCHED FILTER RESPONSE',23,2.0,1)
        CALL MESSAG('TRACKED VEH',11,.25,5.8)
        CALL MESSAG('WHEELED VEH',11,2.00,5.8)
        CALL MESSAG('SHELL BLAST',11,4.00,5.8)
        CALL MESSAG('HELO',4,6.0,5.8)
        CALL MESSAG('PERSONNEL',9,7.55,5.8)
        CALL MESSAG('SENSOR NUMBER',13,.25,.25)
        CALL INTNO(IORR,2.5,.25)
        CALL MESSAG('EVENT NUMBER',12,.25,.45)
        CALL INTNO(TESTNO,2.5,.45)
        CALL MESSAG('TIME PERIOD(SEC)',16,.25,.05)
        TS = FLOAT(IFIX(TIME(1)))
        TF = FLOAT(IFIX(TIME(1024) + 1.))
        CALL REALNO(TS,2,2.4,.05)
```

```
      CALL REALNO(TF,2,3.2,.05)                                         MFI08170
      CALL INTAXS                                                       MFI08180
      CALL YAXANG (0.)                                                  MFI08190
      CALL XTICKS(10)                                                   MFI08200
      CALL YTICKS(10)                                                   MFI08210
      CALL GRAF(0.,2000.0,10240.0,-1.0,0.2,1.0)                         MFI08220
      CALL CURVE(UNIT,RMAT,10240,0.)                                    MFI08230
      CALL DASH                                                         MFI08240
      CALL CURVE (XRAY1,YRAY,2,0)                                       MFI08250
      CALL CURVE (XRAY2,YRAY,2,0)                                       MFI08260
      CALL CURVE (XRAY3,YRAY,2,0)                                       MFI08270
      CALL CURVE (XRAY4,YRAY,2,0)                                       MFI08280
      CALL ENDPL(0)                                                     MFI08290
391   RETURN                                                            MFI08300
      END                                                               MFI08310
C                                                                       MFI08320
C    SUBROUTINE MYDATA SUBTRACTS OFF THE SENSOR BIAS AND CREATES FROM THEMFI08330
C    INTEGER DATA ARRAY PASSED TO IT, REAL AND COMPLEX ARRAYS.          MFI08340
C                                                                       MFI08350
      SUBROUTINE MYDATA(T,DATA,N,RA,A)                                  MFI08360
      REAL    T(N),RA(N)                                                MFI08370
      INTEGER*2 DATA(N)                                                 MFI08380
      COMPLEX A(N)                                                      MFI08390
      DO 10 I=1,N                                                       MFI08400
         IDA=DATA(I)-2048                                               MFI08410
         IF (IDA .GT. 2047 .OR.  IDA .LT. -2048) IDA=0                  MFI08420
         RA(I)=FLOAT(IDA)                                               MFI08430
10    CONTINUE                                                          MFI08440
      SUM=0.0                                                           MFI08450
      DO 20 K=1,N                                                       MFI08460
         AIM=0.0                                                        MFI08470
         A(K)=CMPLX(RA(K),AIM)                                          MFI08480
         SUM=SUM+RA(K)                                                  MFI08490
20    CONTINUE                                                          MFI08500
      AVG=SUM/FLOAT(N)                                                  MFI08510
C                                                                       MFI08520
C   CORRECTION FOR SWITCHING SPIKES IN DATA                             MFI08530
C                                                                       MFI08540
      RA(1)  = 0                                                        MFI08550
      RA(2)  = 0                                                        MFI08560
      RA(3)  = 0                                                        MFI08570
      RA(4)  = 0                                                        MFI08580
      RA(1021) = 0                                                      MFI08590
      RA(1022) = 0                                                      MFI08600
      RA(1023) = 0                                                      MFI08610
      RA(1024) = 0                                                      MFI08620
      RETURN                                                            MFI08630
      END                                                               MFI08640
```

```
C
C     SUBROUTINE MAXMIN FINDS THE MAXIMUM AND MINIMUM VALUES FOR THE ARRAY
C     PASSED TO IT.  IT ALSO RETURNS THE ELEMENT NUMBERS ASSOCIATED WITH
C     THE MAXIMUM AND MINIMUM VALUES.
C
      SUBROUTINE MAXMIN(A,N,K,AMAX,L,AMIN)
      REAL A(N)
      AMAX = A(1)
      AMIN = A(1)
      K=1
      L=1
      DO 10 I=2,N
         IF ((A(I)).LE.(AMAX)) GO TO 20
            AMAX=A(I)
            K=I
20    IF ((A(I)).GE.(AMIN)) GO TO 10
            AMIN=A(I)
            L=I
10    CONTINUE
      IF((ABS(AMIN)).GT.(ABS(AMAX))) AMAX = AMIN
      RETURN
      END
C
C    SUBROUTINE SPCTRM COMPUTES THE NORMALIZED POWER FOR THE COMPLEX ARRAY
C    PASSED TO IT.
C
      SUBROUTINE SPCTRM(T,RA,F,A,N,PWR,PWRDB,PHASE)
      REAL T(N),RA(N),F(N),PWR(N),PHASE(N)
      COMPLEX A(N)
      DO 10 K=1,N
         PWR(K)=CABS(A(K))**2
30    AR=REAL(A(K))
         AI=AIMAG(A(K))
         IF (AR .EQ. 0.0 .AND. AI .EQ. 0.0) GO TO 40
            PHASE(K)=ATAN2(AI,AR)
            GO TO 50
40       PHASE(K)=0.0
50    CONTINUE
10    CONTINUE
      N2=N/2
      CALL MAXMIN(PWR,N,KP,PMAX,LP,PMIN)
C    NORMALIZE POWER SPECTRUM
      DO 60 J=1,N
         PWR(J)=PWR(J)/ABS(PMAX)
60    CONTINUE
      RETURN
      END
C
```

```
C
C     SUBROUTINE MULTI IS THE MULTIPLE TARGET DIRECTION ROUTINE.
C     DIRECTIONS ARE COMPUTED FOR UP TO FIVE TARGET CLASSES.  FOR SINGLE
C     TARGETS, ADAPTIVE MATCHED FILTERING IS PERFORMED ALLOWING FOR
C     IMPROVED ACCURRACY.
C
      SUBROUTINE MULTI(RMAG1,RMAG2,RMAG3,RMAG4,RMAG5,RMAG6,RMAG7,
     1RMAG8,RMAG9,NTYP,SCALE,IORR,THRES,BCAT,TESTNO,JL,C,TS,TF,
     2V,A,ND,NW,NUMSEN,TIME,ITRGET,DIRECT,DIR,DIRC)
      INTEGER NTYP(6),C,NW,ITRGET(9,5),DIRECT,
     1NTR,ND(4),TESTNO,NUMSEN
      REAL RMAG1(1024),RMAG2(1024),RMAG3(1024),RMAG4(1024),RMAG5(1024),
     1RMAG6(1024),RMAG7(1024),RMAG8(1024),RMAG9(1024),
     2BCAT(5120),CCAT(5120),TIME(1024),DIR(5),DIRC,
     3X(9),Y(9),THETA(9),T(9),V(4),A(4),RMAG(1024)
C
      NSEN = 1
      NH = NW/2
      DO 6 I1 = 1,5120
         CCAT(I1) = BCAT(I1)
6     CONTINUE
C
C
C     ADAPTIVE MATCHED FILTERING FOR A SINGLE TARGET IS ACCOMPLISHED
C     BY USING THE SELECTED SENSOR'S SEISMIC DATA AS THE MATCHED FILTER
C     WHEN COMPUTING THE IDENTIFIED TARGET'S DIRECTION.
C
      NTR = 0
      DO 888 I14 = 1,4
         IF((V(I14)).NE.(0)) GO TO 5
888   CONTINUE
      DO 377 JV = 1,5
         IF((NTYP(JV)).NE.(0)) NTR = NTR + 1
377   CONTINUE
      IF((NTYP(1)).EQ.(0)) GO TO 1
      IF((NTR).NE.(1)) GO TO 1
      NTR = NTR + 1
       DO 151 I2 = 1,1024
          CCAT(I2) = RMAG1(I2)
151    CONTINUE
1     IF((NTYP(2)).EQ.(0)) GO TO 2
      IF((NTR).NE.(1)) GO TO 2
      NTR = NTR + 1
       DO 102 I3 = 1,1024
          CCAT(I3 + 1024) = RMAG2(I3)
102    CONTINUE
2     IF((NTYP(3)).EQ.(0)) GO TO 3
      IF((NTR).NE.(1)) GO TO 3
      NTR = NTR + 1
```

```
      DO 103 I4 = 1,1024                                                MFI09610
          CCAT(I4 + 2048) = RMAG3(I4)                                   MFI09620
103    CONTINUE                                                         MFI09630
3     IF((NTYP(4)).EQ.(0)) GO TO 4                                      MFI09640
      IF((NTR).NE.(1)) GO TO 4                                          MFI09650
      NTR = NTR + 1                                                     MFI09660
       DO 104 I5 = 1,1024                                               MFI09670
          CCAT(I5 + 3072) = RMAG4(I5)                                   MFI09680
104    CONTINUE                                                         MFI09690
4     IF((NTYP(5)).EQ.(0)) GO TO 5                                      MFI09700
      IF((NTR).NE.(1)) GO TO 5                                          MFI09710
       DO 105 I6 = 1,1024                                               MFI09720
          CCAT(I6 + 4096) = RMAG5(I6)                                   MFI09730
105    CONTINUE                                                         MFI09740
5     DO 127 J2 = 1,1024                                                MFI09750
        RMAG(J2) = 0.                                                   MFI09760
127   CONTINUE                                                          MFI09770
      II = 0                                                            MFI09780
C                                                                       MFI09790
C     FOR WINDOWING OF DATA, FIND THE POSITION OF THE PEAK AMPLITUDE    MFI09800
C                                                                       MFI09810
      IF((IORR).EQ.(1)) CALL MAXMIN(RMAG1,1024,LM,RMAX,LIM,RMIN)        MFI09820
      IF((IORR).EQ.(2)) CALL MAXMIN(RMAG2,1024,LM,RMAX,LIM,RMIN)        MFI09830
      IF((IORR).EQ.(3)) CALL MAXMIN(RMAG3,1024,LM,RMAX,LIM,RMIN)        MFI09840
      IF((IORR).EQ.(4)) CALL MAXMIN(RMAG4,1024,LM,RMAX,LIM,RMIN)        MFI09850
      IF((IORR).EQ.(5)) CALL MAXMIN(RMAG5,1024,LM,RMAX,LIM,RMIN)        MFI09860
      IF((IORR).EQ.(6)) CALL MAXMIN(RMAG6,1024,LM,RMAX,LIM,RMIN)        MFI09870
      IF((IORR).EQ.(7)) CALL MAXMIN(RMAG7,1024,LM,RMAX,LIM,RMIN)        MFI09880
      IF((IORR).EQ.(8)) CALL MAXMIN(RMAG8,1024,LM,RMAX,LIM,RMIN)        MFI09890
      IF((IORR).EQ.(9)) CALL MAXMIN(RMAG9,1024,LM,RMAX,LIM,RMIN)        MFI09900
C                                                                       MFI09910
C     REPEATED CALLS TO THE MATCHED FILTER ROUTINE ARE MADE WITH WINDOWED MFI09920
C     DATA FROM ALL SENSORS.                                            MFI09930
C                                                                       MFI09940
      DO 555 IX = 1,9                                                   MFI09950
        DO 666 IY = 1,5                                                 MFI09960
        ITRGET(IX,IY) = 0                                               MFI09970
666     CONTINUE                                                        MFI09980
555   CONTINUE                                                          MFI09990
        IF((LM - NH).LE.(0)) LM = NH                                    MFI10000
        IF((LM).GT.(1024 - NH)) LM = 1024 - NH                          MFI10010
      DO 111 I7 = 1,NW                                                  MFI10020
        RMAG(I7) = RMAG1(LM - NH + I7)                                  MFI10030
111   CONTINUE                                                          MFI10040
      NTYP(1) = 0                                                       MFI10050
      NTYP(2) = 0                                                       MFI10060
      NTYP(3) = 0                                                       MFI10070
      NTYP(4) = 0                                                       MFI10080
```

```
      NTYP(5) = 0
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
     1TS,TF,TIME)
      ITRGET(1,1) = NTYP(1)
      ITRGET(1,2) = NTYP(2)
      ITRGET(1,3) = NTYP(3)
      ITRGET(1,4) = NTYP(4)
      ITRGET(1,5) = NTYP(5)
      DO 112 I8 = 1,NW
        RMAG(I8) = RMAG2(LM - NH + I8)
  112 CONTINUE
      NTYP(1) = 0
      NTYP(2) = 0
      NTYP(3) = 0
      NTYP(4) = 0
      NTYP(5) = 0
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
     1TS,TF,TIME)
      ITRGET(2,1) = NTYP(1)
      ITRGET(2,2) = NTYP(2)
      ITRGET(2,3) = NTYP(3)
      ITRGET(2,4) = NTYP(4)
      ITRGET(2,5) = NTYP(5)
      DO 113 I9 = 1,NW
        RMAG(I9) = RMAG3(LM - NH + I9)
  113 CONTINUE
      NTYP(1) = 0
      NTYP(2) = 0
      NTYP(3) = 0
      NTYP(4) = 0
      NTYP(5) = 0
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
     1TS,TF,TIME)
      ITRGET(3,1) = NTYP(1)
      ITRGET(3,2) = NTYP(2)
      ITRGET(3,3) = NTYP(3)
      ITRGET(3,4) = NTYP(4)
      ITRGET(3,5) = NTYP(5)
      DO 114 IA = 1,NW
        RMAG(IA) = RMAG4(LM - NH + IA)
  114 CONTINUE
      NTYP(1) = 0
      NTYP(2) = 0
      NTYP(3) = 0
      NTYP(4) = 0
      NTYP(5) = 0
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
     1TS,TF,TIME)
```

```
      ITRGET(4,1) = NTYP(1)                                             MFI10570
      ITRGET(4,2) = NTYP(2)                                             MFI10580
      ITRGET(4,3) = NTYP(3)                                             MFI10590
      ITRGET(4,4) = NTYP(4)                                             MFI10600
      ITRGET(4,5) = NTYP(5)                                             MFI10610
      DO 115 IB = 1,NW                                                  MFI10620
        RMAG(IB) = RMAG5(LM - NH + IB)                                  MFI10630
  115 CONTINUE                                                          MFI10640
      NTYP(1) = 0                                                       MFI10650
      NTYP(2) = 0                                                       MFI10660
      NTYP(3) = 0                                                       MFI10670
      NTYP(4) = 0                                                       MFI10680
      NTYP(5) = 0                                                       MFI10690
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,   MFI10700
     1TS,TF,TIME)                                                       MFI10710
      ITRGET(5,1) = NTYP(1)                                             MFI10720
      ITRGET(5,2) = NTYP(2)                                             MFI10730
      ITRGET(5,3) = NTYP(3)                                             MFI10740
      ITRGET(5,4) = NTYP(4)                                             MFI10750
      ITRGET(5,5) = NTYP(5)                                             MFI10760
      DO 116 IC = 1,NW                                                  MFI10770
        RMAG(IC) = RMAG6(LM - NH + IC)                                  MFI10780
  116 CONTINUE                                                          MFI10790
      NTYP(1) = 0                                                       MFI10800
      NTYP(2) = 0                                                       MFI10810
      NTYP(3) = 0                                                       MFI10820
      NTYP(4) = 0                                                       MFI10830
      NTYP(5) = 0                                                       MFI10840
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,   MFI10850
     1TS,TF,TIME)                                                       MFI10860
      ITRGET(6,1) = NTYP(1)                                             MFI10870
      ITRGET(6,2) = NTYP(2)                                             MFI10880
      ITRGET(6,3) = NTYP(3)                                             MFI10890
      ITRGET(6,4) = NTYP(4)                                             MFI10900
      ITRGET(6,5) = NTYP(5)                                             MFI10910
      IF((NUMSEN).EQ.(6)) GO TO 101                                     MFI10920
      DO 117 ID = 1,NW                                                  MFI10930
        RMAG(ID) = RMAG7(LM - NH + ID)                                  MFI10940
  117 CONTINUE                                                          MFI10950
      NTYP(1) = 0                                                       MFI10960
      NTYP(2) = 0                                                       MFI10970
      NTYP(3) = 0                                                       MFI10980
      NTYP(4) = 0                                                       MFI10990
      NTYP(5) = 0                                                       MFI11000
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,   MFI11010
     1TS,TF,TIME)                                                       MFI11020
      ITRGET(7,1) = NTYP(1)                                             MFI11030
      ITRGET(7,2) = NTYP(2)                                             MFI11040
```

```
      ITRGET(7,3) = NTYP(3)
      ITRGET(7,4) = NTYP(4)
      ITRGET(7,5) = NTYP(5)
      DO 118 IE = 1,NW
        RMAG(IE) = RMAG8(LM - NH + IE)
118   CONTINUE
      NTYP(1) = 0
      NTYP(2) = 0
      NTYP(3) = 0
      NTYP(4) = 0
      NTYP(5) = 0
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
     1TS,TF,TIME)
      ITRGET(8,1) = NTYP(1)
      ITRGET(8,2) = NTYP(2)
      ITRGET(8,3) = NTYP(3)
      ITRGET(8,4) = NTYP(4)
      ITRGET(8,5) = NTYP(5)
      DO 119 IF = 1,NW
        RMAG(IF) = RMAG9(LM - NH + IF)
119   CONTINUE
      NTYP(1) = 0
      NTYP(2) = 0
      NTYP(3) = 0
      NTYP(4) = 0
      NTYP(5) = 0
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
     1TS,TF,TIME)
      ITRGET(9,1) = NTYP(1)
      ITRGET(9,2) = NTYP(2)
      ITRGET(9,3) = NTYP(3)
      ITRGET(9,4) = NTYP(4)
      ITRGET(9,5) = NTYP(5)
      NTYP(1) = 0
      NTYP(2) = 0
      NTYP(3) = 0
      NTYP(4) = 0
      NTYP(5) = 0
       DO 679 IH = 1,5
         DIR(IH) = 999.
679    CONTINUE
101    RETURN
       END
C
C     MULTIPLE DIRECTION OUTPUT PLOTTING
C
      SUBROUTINE MLTPLT(C,SCALE,TESTNO,TS,TF,DIR,V,A,ND,ITRGET)
      INTEGER C,ND(4),ITRGET(9,5)
```

```
      REAL DIR(5),V(4),A(4)                                              MFI11530
      IF((C).EQ.(1)) CALL COMPRS                                         MFI11540
      IF((C).NE.(1)) CALL TEK618                                         MFI11550
      CALL PAGE(11.0,8.5)                                                MFI11560
      CALL NOBRDR                                                        MFI11570
      CALL BLOWUP(SCALE)                                                 MFI11580
      CALL AREA2D(10.0,6.0)                                              MFI11590
      CALL FRAME                                                         MFI11600
      CALL HEADIN('MULTIPLE TARGET - MATCHED FILTER OUTPUT',39,2.0,1)    MFI11610
      CALL MESSAG('EVENT NUMBER ',13,3.0,5.8)                            MFI11620
      CALL INTNO(TESTNO,5.0,5.8)                                         MFI11630
      CALL MESSAG('TIME PERIOD(SEC)',16,3.0,5.4)                         MFI11640
      CALL REALNO(TS,2,4.9,5.4)                                          MFI11650
      CALL REALNO(TF,2,5.7,5.4)                                          MFI11660
      IF(((ITRGET(1,1)).NE.(0)).AND.((DIR(1)).NE.(999.)))                MFI11670
     1 CALL MESSAG('TRACKED VEHICLE    DIRECTION =',29,3.0,5.0)          MFI11680
      IF(((ITRGET(1,1)).NE.(0)).AND.((DIR(1)).NE.(999.)))                MFI11690
     1 CALL REALNO(DIR(1),2,6.5,5.0)                                     MFI11700
      IF(((ITRGET(1,2)).NE.(0)).AND.((DIR(2)).NE.(999.)))                MFI11710
     1 CALL MESSAG('WHEELED VEHICLE    DIRECTION =',29,3.0,4.6)          MFI11720
      IF(((ITRGET(1,2)).NE.(0)).AND.((DIR(2)).NE.(999.)))                MFI11730
     1 CALL REALNO(DIR(2),2,6.5,4.6)                                     MFI11740
      IF(((ITRGET(1,3)).NE.(0)).AND.((DIR(3)).NE.(999.)))                MFI11750
     1 CALL MESSAG('SHELL BLAST        DIRECTION =',29,3.0,4.2)          MFI11760
      IF(((ITRGET(1,3)).NE.(0)).AND.((DIR(3)).NE.(999.)))                MFI11770
     1 CALL REALNO(DIR(3),2,6.5,4.2)                                     MFI11780
      IF(((ITRGET(1,4)).NE.(0)).AND.((DIR(4)).NE.(999.)))                MFI11790
     1 CALL MESSAG('HELICOPTER         DIRECTION =',29,3.0,3.8)          MFI11800
      IF(((ITRGET(1,4)).NE.(0)).AND.((DIR(4)).NE.(999.)))                MFI11810
     1 CALL REALNO(DIR(4),2,6.5,3.8)                                     MFI11820
      IF(((ITRGET(1,5)).NE.(0)).AND.((DIR(5)).NE.(999.)))                MFI11830
     1 CALL MESSAG('PERSONNEL          DIRECTION =',29,3.0,3.4)          MFI11840
      IF(((ITRGET(1,5)).NE.(0)).AND.((DIR(5)).NE.(999.)))                MFI11850
     1 CALL REALNO(DIR(5),2,6.5,3.4)                                     MFI11860
      CALL MESSAG('SIMULATED TRKD VEHICLE TARGET FREQUENCY',39,3.0,2.8)  MFI11870
      CALL REALNO(V(1),2,8.0,2.8)                                        MFI11880
      CALL MESSAG('AMPLITUDE',9,4.0,2.6)                                 MFI11890
      CALL MESSAG('DIRECTION',9,4.0,2.4)                                 MFI11900
      CALL REALNO(A(1),4,5.5,2.6)                                        MFI11910
      D = FLOAT(ND(1))                                                   MFI11920
      CALL REALNO(D,4,5.5,2.4)                                           MFI11930
      CALL MESSAG('SIMULATED WHLD VEHICLE TARGET FREQUENCY',39,3.0,2.2)  MFI11940
      CALL REALNO(V(2),2,8.0,2.2)                                        MFI11950
      CALL MESSAG('AMPLITUDE',9,4.0,2.0)                                 MFI11960
      CALL MESSAG('DIRECTION',9,4.0,1.80)                                MFI11970
      CALL REALNO(A(2),4,5.5,2.0)                                        MFI11980
      D = FLOAT(ND(2))                                                   MFI11990
      CALL REALNO(D,4,5.5,1.8)                                           MFI12000
```

```
      CALL MESSAG('SIMULATED HELICOPTER TARGET FREQUENCY',37,3.0,1.6)
      CALL REALNC(V(3),2,8.0,1.6)
      CALL MESSAG('AMPLITUDE',9,4.0,1.4)
      CALL MESSAG('DIRECTION',9,4.0,1.2)
      CALL REALNC(A(3),4,5.5,1.4)
      D = FLCAT(NC(3))
      CALL REALNO(D,4,5.5,1.2)
      CALL MESSAG('SIMULATED PERSONNEL TARGET FRECUENCY',36,3.0,1.0)
      CALL REALNO(V(4),2,8.C,1.0)
      CALL MESSAG('AMPLITUDE',9,4.0,0.8)
      CALL MESSAG('DIRECTION',9,4.0,0.6)
      CALL REALNO(A(4),4,5.5,C.8)
      D = FLCAT(NC(4))
      CALL REALNO(D,4,5.5,0.6)
      CALL ENCPL(0)
      RETURN
      END
C
C    SUBRCUTINE RMS CCMPUTES THE ROOT MEAN SQUARE VALUE OF THE N-ARRAY
C    CATA PASSED TC IT
C
      SUBROUTINE RMS(A,N,TMAX)
      REAL A(N)
      TMAX = C.
      DO 200 I = 1,N
         TMAX = (A(I))**2 + TMAX
200   CONTINUE
      TMAX =  SQRT(TMAX)
      RETURN
      END
C
C    SUBROUTINE AVG COMPUTES THE AVERAGE VALUE OF THE N-ARRAY DATA PASSED
C    TO IT
C
      SUBROUTINE AVG(A,N,AAVG)
      REAL A(N)
      AAVG = C.
      DO 100 I = 1,N
         AAVG = A(N) + AAVG
100   CONTINUE
      AAVG = AAVG/FLOAT(N)
      RETURN
      END
C
C    SUBROUTINE SIMULATE ALLOWS FOR THE MCDIFICATICN OF EXPERIMENTAL
C    CATA TO INCLUDE UP TC FOUR SIMULATED TARGETS, ANC MODIFICATION OF
C    SEISMIC SIGNAL AMPLITUDES ABOVE THE SELECTED NOISE LEVEL.
C
```

```
      SUBROUTINE SIMULT(RA1,RA2,RA3,RA4,RA5,RA6,RA7,RA8,RA9,
     1A1,A2,A3,A4,A5,A6,A7,A8,A9,
     2V,A,ND,TNOISE,TIME,JL,P,REDUCE,NUMSEN)
      INTEGER ND(4),NUMSEN
      REAL RA1(1024),RA2(1024),RA3(1024),RA4(1024),RA5(1024),RA6(1024),
     1RA7(1024),RA8(1024),RA9(1024),
     2V(4),A(4),TNOISE,TIME(1024),P(9,4),REDUCE
      COMPLEX A1(1024),A2(1024),A3(1024),A4(1024),A5(1024),
     1A6(1024),A7(1024),A8(1024),A9(1024)
      WRITE(6,10)
   10 FORMAT('0','ENTER THE FOUR SIMULATION FREQUENCIES-REAL-')
      WRITE(6,11)
   11 FORMAT('0','ENTER FREQUENCY')
      READ(10,12)V(1)
      WRITE(6,11)
      READ(10,12)V(2)
      WRITE(6,11)
      READ(10,12)V(3)
      WRITE(6,11)
      READ(10,12)V(4)
   12 FORMAT(F2.0)
      WRITE(6,13)
   13 FORMAT('0','ENTER AMPLITUDES FOR EACH FREQUENCY-REAL-')
      WRITE(6,14)
   14 FORMAT('0','ENTER AMPLITUDE')
      READ(10,15)A(1)
      WRITE(6,14)
      READ(10,15)A(2)
      WRITE(6,14)
      READ(10,15)A(3)
      WRITE(6,14)
      READ(10,15)A(4)
   15 FORMAT(F5.1)
      WRITE(6,16)
      WRITE(6,18)
      WRITE(6,19)
   16 FORMAT('0','ENTER TARGET ANGLE FOR EACH FREQUENCY-I3-')
   18 FORMAT('0','SIX SENSORS ALLOWABLE ANGLES;0,60,120,180,240,300')
   19 FORMAT('0','FOR NINE SENSORS;0,40,80,120,160,200,240,280,320')
      WRITE(6,17)
   17 FORMAT('0','ENTER ANGLE')
      READ(10,55)ND(1)
      WRITE(6,17)
      READ(10,55)ND(2)
      WRITE(6,17)
      READ(10,55)ND(3)
      WRITE(6,17)
      READ(10,55)ND(4)
```

```
55    FORMAT(I3)
      WRITE(6,21)
      REDUCE = 1.
21    FORMAT('0','TO MODIFY AMPLITUDE CF SIGNAL ABOVE NOISE THRESHOLD')
      WRITE(6,22)
22    FORMAT('0','ENTER - 1 -,ELSE ENTER - 2 -')
      READ(10,23)N
23    FORMAT(I1)
      IF((N).NE.(1)) GO TC 26
      WRITE(6,24)
24    FORMAT('0','ENTER NOISE REDUCTION FACTOR, 1 OR GREATER-REAL-')
      READ(10,25)REDUCE
25    FORMAT(F6.1)
26    PI = 3.141593
      PP = 2*PI/360.0
C
C   PHASE DIFFERENCES ARE USED FCR CREATING THE SIMULATED TARGET'S
C   ARRIVAL ANGLE
C
      DO 1 I = 1,4
      IF((NUMSEN).EQ.(9)) GO TO 42
        IF((NE(I)).NE.(0)) GC TO 61
        P(1,I) = C.
        P(2,I) = -45.
        P(3,I) = -135.
        P(4,I) = -180.
        P(5,I) = -135.
        P(6,I) = -45.
61      IF((NE(I)).NE.(60)) GO TO 62
        P(1,I) = -45.
        P(2,I) =  0.
        P(3,I) = -45.
        P(4,I) = -135.
        P(5,I) = -180.
        P(6,I) = -135.
62      IF((NE(I)).NE.(120)) GO TO 63
        P(1,I) = -135.
        P(2,I) = -45.
        P(3,I) =  0.
        P(4,I) = -45.
        P(5,I) = -135.
        P(6,I) = -180.
63      IF((NE(I)).NE.(180)) GO TO 64
        P(1,I) = -180.
        P(2,I) = -135.
        P(3,I) = -45.
        P(4,I) =  0.
        P(5,I) = -45.
```

```
      P(6,I) = -135.                                                     MFI13450
64    IF((NC(I)).NE.(240)) GO TO 65                                      MFI13460
      P(1,I) = -135.                                                     MFI13470
      P(2,I) = -180.                                                     MFI13480
      P(3,I) = -135.                                                     MFI13490
      P(4,I) = -45.                                                      MFI13500
      P(5,I) =  0.                                                       MFI13510
      P(6,I) = -45.                                                      MFI13520
65    IF((NC(I)).NE.(300)) GO TO 42                                      MFI13530
      P(1,I) = -45.                                                      MFI13540
      P(2,I) = -135.                                                     MFI13550
      P(3,I) = -180.                                                     MFI13560
      P(4,I) = -135.                                                     MFI13570
      P(5,I) = -45.                                                      MFI13580
      P(6,I) =  0.                                                       MFI13590
42    IF ((NUMSEN).NE.(9)) GO TO 41                                      MFI13600
      IF((NC(I)).NE.(0)) GO TO 91                                        MFI13610
      P(1,I) =  0.                                                       MFI13620
      P(2,I) = -21.3                                                     MFI13630
      P(3,I) = -76.                                                      MFI13640
      P(4,I) = -138.7                                                    MFI13650
      P(5,I) = -180.                                                     MFI13660
      P(6,I) = -180.                                                     MFI13670
      P(7,I) = -138.7                                                    MFI13680
      P(8,I) = -76.                                                      MFI13690
      P(9,I) = -21.3                                                     MFI13700
91    IF((NC(I)).NE.(40)) GO TO 92                                       MFI13710
      P(1,I) = -21.3                                                     MFI13720
      P(2,I) =  0.                                                       MFI13730
      P(3,I) = -21.3                                                     MFI13740
      P(4,I) = -76.                                                      MFI13750
      P(5,I) = -138.7                                                    MFI13760
      P(6,I) = -180.                                                     MFI13770
      P(7,I) = -180.                                                     MFI13780
      P(8,I) = -138.7                                                    MFI13790
      P(9,I) = -76.                                                      MFI13800
92    IF((NC(I)).NE.(80)) GO TO 93                                       MFI13810
      P(1,I) = -76.                                                      MFI13820
      P(2,I) = -21.3                                                     MFI13830
      P(3,I) =  0.                                                       MFI13840
      P(4,I) = -21.3                                                     MFI13850
      P(5,I) = -76.                                                      MFI13860
      P(6,I) = -138.7                                                    MFI13870
      P(7,I) = -180.                                                     MFI13880
      P(8,I) = -180.                                                     MFI13890
      P(9,I) = -138.7                                                    MFI13900
93    IF((NC(I)).NE.(120)) GO TO 94                                      MFI13910
      P(1,I) = -138.7                                                    MFI13920
```

```
      P(2,I) = -76.                                                     MFI13930
      P(3,I) = -21.3                                                    MFI13940
      P(4,I) =  0.                                                      MFI13950
      P(5,I) = -21.3                                                    MFI13960
      P(6,I) = -76.                                                     MFI13970
      P(7,I) = -138.7                                                   MFI13980
      P(8,I) = -180.                                                    MFI13990
      P(9,I) = -180.                                                    MFI14000
94    IF((NC(I)).NE.(160)) GO TO 95                                     MFI14010
      P(1,I) = -180.                                                    MFI14020
      P(2,I) = -138.7                                                   MFI14030
      P(3,I) = -76.                                                     MFI14040
      P(4,I) = -21.3                                                    MFI14050
      P(5,I) =  0.                                                      MFI14060
      P(6,I) = -21.3                                                    MFI14070
      P(7,I) = -76.                                                     MFI14080
      P(8,I) = -138.7                                                   MFI14090
      P(9,I) = -180.                                                    MFI14100
95    IF((NC(I)).NE.(200)) GO TO 96                                     MFI14110
      P(1,I) = -180.                                                    MFI14120
      P(2,I) = -180.                                                    MFI14130
      P(3,I) = -138.7                                                   MFI14140
      P(4,I) = -76.                                                     MFI14150
      P(5,I) = -21.3                                                    MFI14160
      P(6,I) =  0.                                                      MFI14170
      P(7,I) = -21.3                                                    MFI14180
      P(8,I) = -76.                                                     MFI14190
      P(9,I) = -138.7                                                   MFI14200
96    IF((NC(I)).NE.(240)) GO TO 97                                     MFI14210
      P(1,I) = -138.7                                                   MFI14220
      P(2,I) = -180.                                                    MFI14230
      P(3,I) = -180.                                                    MFI14240
      P(4,I) = -138.7                                                   MFI14250
      P(5,I) = -76.                                                     MFI14260
      P(6,I) = -21.3                                                    MFI14270
      P(7,I) =  0.                                                      MFI14280
      P(8,I) = -21.3                                                    MFI14290
      P(9,I) = -76.                                                     MFI14300
97    IF((NC(I)).NE.(280)) GO TO 98                                     MFI14310
      P(1,I) = -76.                                                     MFI14320
      P(2,I) = -138.7                                                   MFI14330
      P(3,I) = -180.                                                    MFI14340
      P(4,I) = -180.                                                    MFI14350
      P(5,I) = -138.7                                                   MFI14360
      P(6,I) = -76.                                                     MFI14370
      P(7,I) = -21.3                                                    MFI14380
      P(8,I) =  0.                                                      MFI14390
      P(9,I) = -21.3                                                    MFI14400
```

```
98        IF((NC(I)).NE.(320)) GO TO 41
          P(1,I) = -21.3
          P(2,I) = -76.
          P(3,I) = -138.7
          P(4,I) = -180.
          P(5,I) = -180.
          P(6,I) = -138.7
          P(7,I) = -76.
          P(8,I) = -21.3
          P(9,I) =  0.
C
C     ADDITION OF THE SIMULATED SIGNALS TO EACH SENSORS DATA
C
41      DO 2 J = 1,1024
          RA1(J) = RA1(J) + (A(I)*COS(2*PI*V(I)*TIME(J) + PP*P(1,I)))
          RA2(J) = RA2(J) + (A(I)*COS(2*PI*V(I)*TIME(J) + PP*P(2,I)))
          RA3(J) = RA3(J) + (A(I)*COS(2*PI*V(I)*TIME(J) + PP*P(3,I)))
          RA4(J) = RA4(J) + (A(I)*COS(2*PI*V(I)*TIME(J) + PP*P(4,I)))
          RA5(J) = RA5(J) + (A(I)*COS(2*PI*V(I)*TIME(J) + PP*P(5,I)))
          RA6(J) = RA6(J) + (A(I)*COS(2*PI*V(I)*TIME(J) + PP*P(6,I)))
          IF((NUMSEN).EQ.(6)) GO TO 2
          RA7(J) = RA7(J) + (A(I)*COS(2*PI*V(I)*TIME(J) + PP*P(7,I)))
          RA8(J) = RA8(J) + (A(I)*COS(2*PI*V(I)*TIME(J) + PP*P(8,I)))
          RA9(J) = RA9(J) + (A(I)*COS(2*PI*V(I)*TIME(J) + PP*P(9,I)))
2       CONTINUE
1       CONTINUE
C
C     SUPRESSION OR ENHANCEMENT OF SEISMIC SIGNALS ABOVE THE NOISE LEVEL
C
        DO 3 K = 1,1024
          IF((ABS(RA1(K))).GT.(TNOISE)) RA1(K) = RA1(K)/REDUCE
          IF((ABS(RA2(K))).GT.(TNOISE)) RA2(K) = RA2(K)/REDUCE
          IF((ABS(RA3(K))).GT.(TNOISE)) RA3(K) = RA3(K)/REDUCE
          IF((ABS(RA4(K))).GT.(TNOISE)) RA4(K) = RA4(K)/REDUCE
          IF((ABS(RA5(K))).GT.(TNOISE)) RA5(K) = RA5(K)/REDUCE
          IF((ABS(RA6(K))).GT.(TNOISE)) RA6(K) = RA6(K)/REDUCE
          IF((NUMSEN).EQ.(6)) GO TO 3
          IF((ABS(RA7(K))).GT.(TNOISE)) RA7(K) = RA7(K)/REDUCE
          IF((ABS(RA8(K))).GT.(TNOISE)) RA8(K) = RA8(K)/REDUCE
          IF((ABS(RA9(K))).GT.(TNOISE)) RA9(K) = RA9(K)/REDUCE
3       CONTINUE
        DO 20 K1 = 1,1024
          ABC = 0.
          A1(K1) = CMPLX(RA1(K1),ABC)
          A2(K1) = CMPLX(RA2(K1),ABC)
          A3(K1) = CMPLX(RA3(K1),ABC)
          A4(K1) = CMPLX(RA4(K1),ABC)
          A5(K1) = CMPLX(RA5(K1),ABC)
```

```
      A6(K1) = CMPLX(RA6(K1),ABC)
      IF((NUMSEN).EQ.(6)) GO TO 20
      A7(K1) = CMPLX(RA7(K1),ABC)
      A8(K1) = CMPLX(RA8(K1),ABC)
      A9(K1) = CMPLX(RA9(K1),ABC)
20    CONTINUE
30    RETURN
      END
C
C   LEAST MEAN SQUARES POLYNOMIAL CURVE FITTING ALGORITHM
C
      SUBROUTINE LMS (NTAR,DIRC,NUMSEN,C,SCALE,TESTNC,TIME,NFRM,Y,
     1VX,X,TT,INDEX,NCR,TRIX,B,WKAREA)
      DIMENSION TRIX(9,9),B(9),WKAREA(108)
      INTEGER NTAR(9),T(9),C,NHALF,MIN,NUMSEN,TESTNC,NFRM
      REAL P,FMIN,S,XS(16),X(9),A(9),Y(360),
     1VX(360),TT(9),TIME(1024),DIRC,SCALE,C
C
C   FIND FIRST SENSOR TO MATCH THE INPUT SIGNAL
C
        MIN = NTAR(1)
        MARK = 1
        IF((NTAR(1)).LT.(1)) GO TO 66
        DO 97 I17 =1,9
            T(I17) = 0
            TT(I17) = 0.
            WRITE(6,53) NTAR(I17)
97     CONTINUE
53     FORMAT(I7)
        DO 11 I2 = 2,NUMSEN
          IF((NTAR(I2)).GE.(MIN)) GO TO 11
          MIN = NTAR(I2)
          MARK = I2
11      CONTINUE
      DO 46 I8 = 1,360
        Y(I8) = C.
        VX(I8) = 0.
46    CONTINUE
C
C   CENTER THE PARABOLA
C
      IF((MIN).EQ.(0)) GO TO 66
      DO 21 K1 = 1,NUMSEN
        INA = 1
        IF((NUMSEN).EQ.(6)) INA = 0
        N = NUMSEN/2 - MARK + K1 + INA   .
        IF((N).GT.(NUMSEN)) N = N - NUMSEN
        IF((N).LT.(1)) N = N + NUMSEN
```

```
      T(N) = NTAR(K1)                                                   MFI15370
21    CONTINUE                                                          MFI15380
      DO 41 L = 1,NUMSEN                                                MFI15390
         T(L) = T(L) - MIN                                              MFI15400
         TT(L) = FLOAT(T(L))                                            MFI15410
41    CONTINUE                                                          MFI15420
      DO 10 I3 = 1,16                                                   MFI15430
         XS(I3) = 0.                                                    MFI15440
10    CONTINUE                                                          MFI15450
      X(1) = 0.                                                         MFI15460
      LN = NUMSEN - 1                                                   MFI15470
      DO 130 I4 = 1,LN                                                  MFI15480
         IF((NUMSEN).EQ.(6)) X(I4 + 1) = (FLOAT(I4*60))/360.            MFI15490
         IF((NUMSEN).EQ.(9)) X(I4 + 1) = (FLOAT(I4*40))/320.            MFI15500
130   CONTINUE                                                          MFI15510
C                                                                       MFI15520
C   LOAD THE MATRICES FOR THE LEAST MEAN SQUARES POLYNOMIAL SOLUTION    MFI15530
C                                                                       MFI15540
      IF(NUMSEN.EQ.6) X(7) =0                                           MFI15550
      IF(NUMSEN.EQ.6) X(8) =0                                           MFI15560
      IF(NUMSEN.EQ.6) X(9) =0                                           MFI15570
      NTC = 2*NUMSEN - 2                                                MFI15580
      DO 20 N1 = 1,NTC                                                  MFI15590
         DO 30 K3 = 1,NUMSEN                                            MFI15600
            XS(N1) = X(K3)**N1 + XS(N1)                                 MFI15610
30    CONTINUE                                                          MFI15620
20    CONTINUE                                                          MFI15630
      DO 47 I2A = 1,9                                                   MFI15640
         DO 49 I3A = 1,9                                                MFI15650
          TRIX(I2A,I3A) = 0.                                            MFI15660
49       CONTINUE                                                       MFI15670
47    CONTINUE                                                          MFI15680
      TRIX(1,1) = NUMSEN                                                MFI15690
      DO 50 N2 = 2,NUMSEN                                               MFI15700
       TRIX(1,N2) = XS(N2 - 1)                                          MFI15710
50    CONTINUE                                                          MFI15720
      DO 40 I5 = 1,NUMSEN                                               MFI15730
        TRIX(2,I5) = XS(I5)                                             MFI15740
        TRIX(3,I5) = XS(I5 + 1)                                         MFI15750
        TRIX(4,I5) = XS(I5 + 2)                                         MFI15760
        TRIX(5,I5) = XS(I5 + 3)                                         MFI15770
        TRIX(6,I5) = XS(I5 + 4)                                         MFI15780
        IF((NUMSEN).EQ.(6)) GO TO 40                                    MFI15790
        TRIX(7,I5) = XS(I5 + 5)                                         MFI15800
        TRIX(8,I5) = XS(I5 + 6)                                         MFI15810
        TRIX(9,I5) = XS(I5 + 7)                                         MFI15820
40    CONTINUE                                                          MFI15830
      DO 60 I6 = 1,9                                                    MFI15840
```

```
         B(I6) = 0
60    CONTINUE
      DO 70 I7 = 2,NUMSEN
         DO 80 K4 = 1,NUMSEN
           B(I7) = B(I7) + (X(K4)**(I7 - 1))*(FLOAT(T(K4)))
80       CONTINUE
70    CONTINUE
      DO 90 I8 = 1,NUMSEN
         B(1) = B(1) +(FLOAT(T(I8)))
90    CONTINUE
      WRITE(6,1)
      WRITE(6,2)
1     FORMAT('0','ENTER DEGREE OF POLYNOMIAL DESIRED -I1-')
2     FORMAT('0','FOR SIX SENSORS ENTER 2 - 4, FOR NINE ENTER 2 - 7')
      READ(10,3) NOR
3     FORMAT(I1)
      NZ = NOR + 1
C
C   CALL THE MATRIX SOLVER LEQT2F VIA SUBROUTINE SOLV TO FIND THE
C   COEFFICIENTS FOR THE LEAST MEAN SQUARES POLYNOMIAL
C
      CALL SOLV (TRIX,NZ,B)
         PMIN = 1E20
         MINI = 1
         DIVIDE = 40.
         IF((NUMSEN).EQ.(6)) DIVIDE = 60.
      NO = NOR + 2
       DO 110 I9 = NO,9
        B(I9) = 0
110   CONTINUE
      IF((NUMSEN).EQ.(6)) INDEX = 360
      IF((NUMSEN).EQ.(9)) INDEX = 320
C
C   CALCULATE PLOTTING DATA FROM THE LEAST MEAN SQUARES POLYNOMIAL
C
      DO 120 NR = 1,INDEX
      Q = FLOAT(NR)/FLOAT(INDEX)
         P = B(1) + B(2)*(Q) + B(3)*(Q**2)
     1 + B(4)*(Q**3)
     2 + B(5)*(Q**4)
     3 + B(6)*(Q**5)
     4 + B(7)*(Q**6)
     5 + B(8)*(Q**7)
     6 + B(9)*(Q**8)
      VX(NR) = FLOAT(NR) + FLOAT(MARK - NUMSEN/2 - INA)*DIVIDE -1.
       Y(NR) = P
      IF((P).GT.(PMIN)) GO TO 120
      MINI = NR
```

```
      FMIN = F
120   CONTINLE
      NHALF = NUMSEN/2
      DIRC = MINI + (MARK - NUMSEN/2 - INA)*CIVIDE
      DO 77 IB = 1,NUMSEN
       X(IB) = X(IB)*FLOAT(INDEX) + FLOAT(MARK - NUMSEN/2 -INA)*DIVIDE
77    CONTINLE
66    RETURN
      ENC
C
C   OUTPUT PLCTTING FCR THE LEAST MEAN SQUARES SUBRCUTINE
C
      SUBROUTINE PLT (C,SCALE,TESTNO,NUMSEN,TIME,YY,VXR,BX,TTB,INCEX,
     1NOR,DIRC,NFRM)
      INTEGER C,TESTNC,NUMSEN,INCEX,NCR,NFRM
      REAL YY(INDEX),BX(NUMSEN),
     1VXR(INCEX),TTB(NUMSEN),TIME(1024),SCALE
      IF((C).EC.(1)) CALL CCMPRS
      IF((C).NE.(1)) CALL TEK618
      CALL PACE(11.0,8.5)
      CALL NCERDR
      CALL BLCWUP(SCALE)
      CALL AREA2C(9.0,6.C)
      CALL FRAME
      CALL XNAME('ANGLE-CEGREES-',14)
      CALL YNAME('RELATIVE DELAY TIMES',20)
      CALL HEACIN('LEAST MEAN SQUARES POLYNOMIAL',29,2.0,1)
      CALL MESSAG('EVENT NUMBER',12,.25,.65)
      CALL INTNO(TESTNO,2.5,.65)
      CALL MESSAG('DEGREE PCLYNOMIAL',17,.25,.45)
      CALL INTNO(NOR,2.5,.45)
C
      IF((NFRM).EC.(0))CALL MESSAG('INITIAL CIRECTION',17,.25,.25)
      IF((NFRM).EC.(0))CALL REALNO(DIRC,2,2.4,.25)
      IF((NFRM).NE.(0))CALL MESSAG('TARGET TYPE',11,.25,.25)
      IF((NFRM).EC.(1))CALL MESSAG('TRK VEH',7,2.5,.25)
      IF((NFRM).EQ.(2))CALL MESSAG('WHLD VEH',8,2.5,.25)
      IF((NFRM).EC.(3))CALL MESSAG('SHL BLST',8,2.5,.25)
      IF((NFRM).EQ.(4))CALL MESSAG('HELO',4,2.5,.25)
      IF((NFRM).EQ.(5))CALL MESSAG('PERS',4,2.5,.25)
      CALL MESSAG('TIME PERICD(SEC)',16,.25,.05)
      S = FLCAT(IFIX(TIME(1)))
      F = FLCAT(IFIX(TIME(1024) + 1.))
      CALL REALNO(S,2,2.4,.05)
      CALL REALNO(F,2,3.2,.05)
      CALL MESSAG('ACTUAL SENSOR DATA ARE INCICATED',32,2.5,5.7)
      CALL MESSAG('BY MARKER SYMBOLS',17,3.3,5.5)
      CALL INTAXS
```

```
      CALL YAXANG(0.)                                                   MFI16810
      CALL XTICKS(10)                                                   MFI16820
      CALL YTICKS(10)                                                   MFI16830
      TS = VXR(1)                                                       MFI16840
      TF = VXR(INDEX)                                                   MFI16850
      CALL MAXMIN(YY,INDEX,K,YMAX,L,YMIN)                               MFI16860
      IF((YMAX).LE.(YMIN)) YMAX = - YMAX                                MFI16870
      CALL MAXMIN(TTB,NUMSEN,K,TMAX,L,TMIN)                             MFI16880
      IF((TMAX).LE.(TMIN)) TMAX = - TMIN                                MFI16890
      IF((TMAX).GT.(YMAX)) YMAX = TMAX                                  MFI16900
      IF((TMIN).LT.(YMIN)) YMIN = TMIN                                  MFI16910
      CALL GRAF(TS,'SCALE',TF,YMIN,'SCALE',YMAX)                        MFI16920
      CALL RESET ('MARKER')                                             MFI16930
      CALL CURVE(VXR,YY,INDEX,0.)                                       MFI16940
      CALL MARKER(10)                                                   MFI16950
      CALL CURVE (BX,TTB,NUMSEN,-1)                                     MFI16960
      CALL RESET ('MARKER')                                             MFI16970
      CALL ENDPL (0)                                                    MFI16980
      RETURN                                                            MFI16990
      END                                                               MFI17000
C                                                                       MFI17010
C     MATRIX SOLVING ROUTINE                                            MFI17020
C                                                                       MFI17030
      SUBROUTINE SOLV(TRIX,NZ,B)                                        MFI17040
      DIMENSION TRIX(9,9),B(9),WKAREA(108)                              MFI17050
      CALL LEQT2F (TRIX,1,NZ,9,B,0,WKAREA,IER)                          MFI17060
      RETURN                                                            MFI17070
      END                                                               MFI17080
```

LIST OF REFERENCES

1. Naval Ocean Systems Command Technical Report 747, Seismic/Acoustic Field Measurements and Analysis, by R. G. Rock, J G. Hoffman, A. T. Roome, L. H. Bossert and J. H. Brown, 9 November 1981.

2. Grant, F. S., Interpretation Theory in Applied Geophysics, pp.5 - 11, McGraw Hill, 1965.

3. Orincon Corporation, Signal Processing for Seismic Surveillance, by H. W. Sorenson, pp.1 - 35, July 1981.

4. Shelef, S. C., Phase Difference/Time Delay Trace Functions and their Application to Bearing Estimation in Arrays, Ph.d. Thesis, Naval Postgraduate School, 1979.

5. Dublin, C. P., Marine Corps Passive Surveillance Requirements, Masters Thesis, Naval Postgraduate School, Monterey, 1980.

6. Skolnik, M. I., Introduction to Radar Systems. pp. 373 - 376, 428 - 431, McGraw-Hill, 1980.

7. Cooper, G. R., and Mc Gillem C. D., Probabilistic Methods of Signal and System Analysis, pp 205 - 210, Holt, Rinehart, and Winston, 1971.

8. Oppenheim, A. V., and Schafer, R. W., Digital Signal Processing, pp. 111 - 115, Prentice - Hall, 1975.

9. Park, S., Detection, Classification and Locating via Seismic/Acoustic Sensors, Masters Thesis, Naval Postgraduate School, Monterey, 1982.

10. Gerald, C. F., Applied Numerical Analysis, pp. 465 - 474, Addision-Wesley, 1980.

11. Salvadori, M. G., and Baron, M. L., Numerical Methods in Engineering, pp.65 - 87, Prentice-Hall, 1961.

12. Naval Postgraduate School Technical Note MVS - 01, Users Guide to MVS at NPS, by June Favorite, pp.1 - 154, July 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Commanding Officer Naval Ocean System Command Code: 8105 Attn: Dan Leonard San Diego, California 92152	3
4.	Professor H. A. Titus, code 62TS Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	12
5.	GTE Systems Attn: Bruce P. Marion P.O. Box 7188 100 Ferguson Drive Mountain View, California 94039	1
6.	Lt. Constantinos Manikas 428 Delavina #122 Monterey, California 93940	1
7.	Commander JITF, JINTACCS Attn: Captain Robert R. Logan Ft Monmouth, New Jersey 07703	1
8.	Ken Duval 250 Polaris San Jose, California 95126	1
9.	Captain James L. Johnston USMC 17410 N.E. 36th St. Redmond, Washington 98052	6

Johnston
Multiple target identification and direction finding using matched filtering techniques.

206664
Thesis
J6673
c.1
Johnston
Multiple target identification and direction finding using matched filtering techniques.